

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ
СІКОРСЬКОГО»**

Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

"На правах рукопису"
УДК 004.02

«До захисту допущено»

Завідувач кафедри

(підпис) О.В. Коваль
(ініціали, прізвище)

“ ____ ” _____ 2019 р.

Магістерська дисертація

зі спеціальності 121 Інженерія програмного забезпечення
за спеціалізацією Інженерія програмного забезпечення розподілених систем
на тему “Підсистема визначення позиції в просторі для мобільного
додатку навігації з використанням доповненої реальності”

Виконав: студент 6 курсу, групи _ТВ-81мп_

Харабар Владислав Володимирович

(прізвище, ім'я, по батькові)

(підпис)

Науковий керівник доц., к.т.н. Гагарін О. О.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Рецензент _____

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних
посилань.

Студент _____

(підпис)

Київ — 2019

Національний технічний університет України
“Київський політехнічний інститут ім. Ігоря Сікорського”

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти другий, магістерський

зі спеціальності — 121 Інженерія програмного забезпечення

за спеціалізацією — Інженерія програмного забезпечення розподілених систем

ЗАТВЕРДЖУЮ
Завідувач кафедри
Коваль О.В.
(прізвище, ініціали) (підпис)
« » 2019 р.

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ ДИСЕРТАЦІЮ СТУДЕНТУ

Харабару Владиславу Володимировичу

(прізвище, ім'я, по батькові)

1. Тема дисертації «Підсистема визначення позиції в просторі для мобільного додатку навігації з використанням доповненої реальності»

Науковий керівник доц., к.т.н., Гагарін Олександр Олександрович

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “04” листопада 2019 року № 3812-с

2. Строк подання студентом дисертації 9 грудня 2019 р.

3. Об'єкт дослідження засоби визначення позиції в приміщеннях на основі комп'ютерного зору

4. Предмет дослідження візуально-інерційна система визначення позиції для потреб внутрішньої навігації в смартфонах

5. Перелік питань, які потрібно розробити Проаналізувати методи визначення позиції в приміщеннях. Розробити сервіс внутрішнього позиціонування на основі вибраного підходу. Інтегрувати підсистему з іншими складовими частинами програмного продукту. Протестувати додаток.

6. Орієнтований перелік ілюстративного матеріалу «Актуальність теми дослідження», «Складові процесу навігації», «Мета та завдання роботи», «Об'єкт, предмет дослідження», «Взаємодія модулів додатку», «Порівняння методів позиціонування», «Запропонований підхід», «Потік даних», «Метод SLAM», «Маркери ArUco», «Принцип синхронізації», «Структура системи», «Алгоритм роботи з програмою», «Демонстрація роботи», «Висновки»

7. Орієнтований перелік публікацій _____
Використання техніки Structure from Motion в системі навігації (XVII міжнародна науково-практична конференція молодих вчених та студентів «Сучасні проблеми наукового забезпечення енергетики»)

8. Дата видачі завдання « 28 » вересня 2018 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання магістерської дисертації	Строки виконання етапів магістерської дисертації	Примітка
1	Отримання завдання	28.09.18 р.	
2	Опрацювання літературних джерел	01.10.18 р. — 03.02.19 р.	
3	Підготовка матеріалів дисертації	04.02 — 31.05.19 р.	
4	Підготовка доповідей на конференції	11.03 — 29.03.19 р.	
6	Розробка програмного продукту	03.06 — 25.10.19 р.	
5	Переддипломна практика	02.09 — 25.10.19 р.	
7	Захист програмного продукту	26.10.19 р.	
8	Розробка стартап-проекту	11.11 — 19.11.19 р.	
9	Передзахист	20.11.19 р.	
10	Оформлення дисертації	21.11- 29.11.19 р.	
11	Захист	16.12.19 р.	

Студент

(підпис)

Харабар В. В.

(прізвище та ініціали)

Науковий керівник

(підпис)

Гагарін О. О.

(прізвище та ініціали)

РЕФЕРАТ

Актуальність теми

Наразі сервіси навігації стали невід'ємними супутниками нашого життя. Переважна більшість мобільних пристроїв підтримує глобальне позиціонування GPS. Всередині ж будівель його застосування не є можливим. Потрібно розробляти специфічні технології, що часто потребують модифікованих мобільних пристроїв.

На ринку внутрішнього позиціонування зараз основним підходом є використання BLE-маяків та сигналів наявних точок доступу Wi-Fi. Проте обидва методи вимагають значних вкладень грошових та трудових ресурсів.

Сучасні смартфони досягли такого рівня продуктивності, що можуть в реальному часі аналізувати зображення із камери та на основі нього визначати відносне переміщення пристрою.

Поєднання вказаних факторів робить розробку оптичних систем позиціонування мобільних пристроїв надзвичайно актуальною.

Метою дослідження є створення підсистеми позиціонування для мобільного додатку внутрішньої навігації, який дозволяє знаходити шлях до точок інтересу користувача та супроводжує його шляхом надання інтерактивних інструкцій в режимі доповненої реальності.

Для досягнення поставленої мети потрібно виконати наступні **задачі дослідження**:

- розглянути, проаналізувати та порівняти методи визначення позиції, зокрема в приміщеннях;
- розробити сервіс внутрішнього позиціонування на основі вибраного підходу;
- інтегрувати підсистему з іншими складовими частинами програмного продукту.

Об'єкт дослідження: Засоби визначення позиції в приміщеннях на основі комп'ютерного зору.

Предмет дослідження: Візуально-інерційна система визначення позиції для потреб внутрішньої навігації в смартфонах.

Методи дослідження: Для розв’язання поставлених задач використовуються методи проектування архітектури програмних систем, теорія моделювання потоків даних, об’єктно-орієнтованого аналізу предметної області.

Наукова новизна:

Удосконалено систему внутрішнього позиціонування через поєднання візуально-інерційної одометрії з синтетичними маркерами, що призвело до можливості розробки системи локалізації без розміщення дорогих пристроїв приймання чи передачі сигналів.

Практичне значення полягає в тому, що розроблений додаток можна застосовувати для потреб пошуку аудиторій в навчальних будівлях, а також для анотації предметів в них. Функціонал додатку дозволяє замінити схеми приміщення для перевикористання в інших галузях.

Апробація: Результати досліджень обговорювались на XVII науково-практичній конференції молодих вчених та студентів «Сучасні проблеми наукового забезпечення енергетики», що проходила в Києві 23-26 квітня 2019 року.

Структура і обсяг дипломної роботи

Магістерська дисертація складається зі вступу, п’яти розділів та висновків, переліку посилань з 46 джерел; включає 25 рисунків, 25 таблиць та 2 додатки. Повний обсяг роботи складає 112 сторінок, з них 5 — перелік посилань та додатки — 13 сторінок.

У вступі обґрунтовано необхідність розробки нових підходів до навігації, задано мету дослідження, визначено завдання розробки.

У першому розділі розглянуте поняття навігації. Була описана практична важливість систем внутрішнього позиціонування та їх потенційні застосування. Визначено місце підсистеми в комплексному додатку та потік даних в ній.

У другому розділі досліджені основні техніки та технології визначення позиції для внутрішньої навігації. Особливу увагу зосереджено на методі SLAM. Проаналізовані існуючі програмні системи, їх переваги та недоліки. Запропонований підхід, що поєднує SLAM та синтетичні маркери. Обґрунтований вибір маркерів ArUco.

Третій розділ присвячено програмній реалізації підсистеми. Описані засоби розробки, що використовуються, налаштування та структура розробленого продукту.

Четвертий розділ наводить інструкції щодо розгортання, інсталяції та користування комплексним програмним продуктом.

У п'ятому розділі надається аналіз можливостей впровадження розробленого продукту в якості стартап-проекту. Визначені стратегії розвитку та маркетингу, проведено SWOT- та фінансовий аналіз, описані фактори ризиків та можливостей.

У висновках викладені отримані результати виконаної роботи та рекомендації щодо подальших досліджень даної теми.

Додатки містять програмний код розробленої підсистеми та тези доповіді конференції, на якій апробувались матеріали дисертації.

Ключові слова: SLAM, СИНТЕТИЧНІ МАРКЕРИ, ARUCO, ВНУТРІШНЯ НАВІГАЦІЯ, ПОЗИЦІОНУВАННЯ, ARCORE.

ABSTRACT

Relevance of the topic

Nowadays navigation services have become an integral part of our lives. The vast majority of mobile devices support global GPS positioning. However, it is not possible to use it inside buildings. In these cases, specific technologies that often require modified mobile devices have to be developed.

On the IPIN market, the main approach now is to use BLE beacons and signals from existing Wi-Fi access points. However, both methods require considerable investment of money and time.

Today's smartphones have reached a level of performance that supports real-time camera image stream processing and determining the relative movement of the device.

The combination of these factors makes the development of optical positioning systems for mobile devices extremely relevant.

The purpose of research is the creation of a positioning subsystem for the mobile indoor navigation application, which helps users to find a way to the points of interest and accompanies to them by providing interactive instructions using augmented reality.

To achieve this goal, we need to complete the following **research objectives**:

- to consider, analyze and compare methods of indoor position determination;
- to develop an indoor positioning service based on the chosen approach;
- integrate the subsystem with other components of the software product.

Object of study: Indoor positioning methods based on computer vision.

Subject of study: A visual-inertial indoor positioning system for navigation using smartphones.

Research methods: To solve the problem, we use software architecture design methods, theory of data flows modeling, object-oriented analysis of the domain.

Scientific novelty:

Internal positioning system has been improved by combining visual-inertial odometry with synthetic markers, which has led to the possibility of developing a localization system without deploying expensive devices for receiving or transmitting signals.

Practical meaning is that the developed application can be used for the purposes of finding rooms in educational buildings, as well as for object annotation. The functionality of the application allows you to replace the map of the facility to reuse in other fields.

Approbation. The results of the research were discussed at the 17th Scientific and Practical Conference of Young Scientists and Students "Modern Problems of Scientific Support for Energy" held in Kyiv, April 23-26, 2019.

Structure and volume of the thesis

The master's thesis consists of an introduction, five sections and conclusions, a list of references with 46 sources; includes 25 figures, 25 tables and 2 appendices. The full amount of work is 112 pages, 5 of them — references list and appendices — 13 pages.

The introduction substantiates the need to develop new approaches to navigation, sets a research goal, and defines the task of development.

The first section describes the concept of navigation. It describes the practical importance of internal positioning systems and their potential applications. Determines place of the subsystem in the resulting application and its data flow.

The second section explores the basic techniques and technologies for indoor navigation positioning. Attention is especially focused on the SLAM method. Existing software systems, their advantages and disadvantages are analyzed. An approach combining SLAM and synthetic markers is proposed. Justified the choice of ArUco markers.

The third section is devoted to the software implementation of the subsystem. We describe used development tools, structure and configuration of the developed product.

The fourth section provides instructions on how to deploy, install, and use the software product — mobile application.

The fifth section provides an analysis of possibilities to implement the product as a startup project. We identified development and marketing strategies, conducted a SWOT- and financial analysis and described risk and opportunity factors.

The conclusions set out the results of the work and provide recommendations for further research on the topic.

The appendices contain the program code of the developed subsystem and the abstracts for the conference where the dissertation materials were discussed.

Keywords: SLAM, SYNTETIC MARKERS, ARUCO, INDOOR NAVIGATION, POSITIONING, ARCORE.

ЗМІСТ

Перелік скорочень, умовних позначень і термінів	11
Вступ.....	12
1 Задача визначення позиції пристроїв у приміщеннях	14
1.1 Задача навігації	14
1.2 Призначення програмного продукту	15
1.3 Вхідні дані	18
1.4 Компоненти системи	19
Висновки до розділу	19
2 Аналіз проблеми визначення позиції мобільного пристрою	20
2.1 Система внутрішнього позиціонування	20
2.1.1 Особливості внутрішнього позиціонування	20
2.1.2 Основні принципи визначення позиції	21
2.1.3 Технології позиціонування	26
2.2 Метод одночасної локалізації та картографування SLAM	31
2.2.1 Розвиток SLAM.....	32
2.2.2 Постановка задачі SLAM.....	33
2.2.3 Зовнішній модуль SLAM та його задачі	36
2.2.4 Типи помилок та способи боротьби з ними.....	37
2.2.5 Моделі представлення оточення	39
2.3 Аналіз існуючих програмних систем	42
2.3.1 Системи зі спеціальними пристроями.....	42
2.3.2 Додатки з використанням візуального визначення позиції	45
2.4 Обраний підхід до підсистеми визначення положення	46
Висновки до розділу	49

3 Розробка програмного продукту.....	50
3.1 Засоби розробки.....	50
3.1.1 Редактор Unity.....	50
3.1.2 Платформа ARCore.....	51
3.2 Структура додатку	53
3.3 Об'єкт сесії ARCore.....	54
3.4 Відстеження позиції	58
3.5 Генерація та розпізнавання маркерів.....	61
3.6 Навігація між поверхами.....	66
Висновки до розділу.....	68
4 Робота користувача з програмною системою	69
4.1 Системні вимоги та інсталяція	69
4.2 Сценарій роботи користувача з системою	70
Висновки до розділу.....	72
5 Розробка стартап-проекту.....	73
5.1 Опис ідеї продукту.....	73
5.2 Технологічний аудит	75
5.3 Аналіз ринкових можливостей.....	76
5.4 Розробка ринкової стратегії.....	85
5.5 Створення маркетингової програми	88
Висновки до розділу.....	91
Висновки	93
Список використаних джерел	95
Додаток А	100
Додаток Б.....	109

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ

AR, ДР — Augmented Reality — доповнена реальність.

BLE — Bluetooth Low Energy — специфікація бездротової технології обміну даними з низьким споживанням електроенергії.

EKF — Extended Kalman Filter — розширений фільтр Калмана.

GPS — Global Positioning System — система глобального позиціонування.

IPIN — Indoor Navigation and Indoor Positioning — внутрішня навігація та позиціонування; часто застосовується відносно ринку.

IPS, СВП — Indoor Navigation System — система внутрішнього позиціонування.

QR-код — QR code — код швидкої відповіді.

SLAM — Simultaneous Location And Mapping — одночасна локалізація та картографування.

SfM — Structure from Motion — метод відновлення структури з руху.

МАЙ — максимум апостеріорної ймовірності.

ВСТУП

Розвиток мобільних пристроїв зробив обчислення більш доступними, ближчими та контекстно-усвідомленими. Смартфони стали незамінними помічниками, зокрема в задачах навігації.

Проте, такі мобільні пристрої ефективно функціонують тільки в ідеальних умовах та середовищі. Наприклад, завантаження програм та іншого контенту можливо лише за наявності покриття мережі чи Wi-Fi. Аналогічно, можливість використання смартфона для навігації у відкритому середовищі (англ. outdoor) напряду залежить від неперервного доступу до сигналів супутників.

Всередині будівель, де відсутні сигнали глобальних систем позиціонування, користувачі не зможуть користуватись звичними функціями мобільного пристрою. Наразі стало зрозумілим, що не існує єдиного ефективного рішення задачі внутрішньої навігації, що засновується на одній технології. Сучасні технології не можуть надати дешевий спосіб глобального позиціонування з точністю в 1 м. Системи потребують встановлення виділеної інфраструктури та наявності в користувачів модифікованих мобільних пристроїв.

Саме тому є актуальною проблема створення дешевої системи навігації всередині будівель, що на покладається на зовнішні мережі та є повністю чи частково автономною.

Наразі, для кожної конкретної спроби впровадження такої системи необхідно окремо аналізувати вимоги, щоб підібрати відповідне рішення. Тому є важливою оцінка параметрів технологій внутрішньої навігації. Кількість таких параметрів є досить значною (точність, покриття, цілісність, доступність, затримка, витрати, приватність, адаптивність). Існує багато технологій позиціонування мобільних пристроїв. На найвищому рівні їх можна поділити на категорії по використовуваним фізичним принципам: інерційна навігація (акселерометри та гіроскопи), на основі механічних хвиль (ультразвук, чутний звук) та з застосуванням електромагнітних хвиль (інфрачервоний, видимий, радіо та ультрахвильовий спектри). Найчастіше базуються на властивостях електромагнітних хвиль радіо [1].

Існуючі рішення часто потребують розгортання дорогої та складної інфраструктури, мережі пристроїв рівномірно розміщених по всій зоні покриття. Виникає питання: «Чи можна досягти точного відстеження переміщення мобільного пристрою без інфраструктури?». Дослідження в цій області йдуть вже досить довго, але тільки нещодавно смартфони набули необхідних потужностей для забезпечення роботи алгоритмів комп'ютерного зору в реальному часі. До цього доводилось використовувати інші методи або делегувати обробку потужним серверам, що створювало додаткові складнощі та збільшувало час відклику системи.

У даній роботі основну увагу зосереджено на використанні систем комп'ютерного зору, що на основі зображення з камери оцінюють положення смартфона.

Основною проблемою роботи є створення підсистеми позиціювання для мобільного додатку, який надає можливість орієнтуватись у внутрішньому просторі будівель без встановлення будь-яких інших пристроїв. Її розв'язком стало застосування синтетичних маркерів для початкової прив'язки пристрою до схеми та візуально-інерційного навігаційного обчислення .

Перший розділ дисертації ставить задачу, що вирішується. В наступному розділі проводиться її аналіз, пошук існуючих рішень, вирізняється концепція підсистеми. Третій розділ повністю присвячений розробці програмного продукту, в ньому розглядаються засоби розробки, доступні технології, особливості їх застосування, алгоритми роботи підсистеми та схеми взаємодії компонентів. В останньому розділі оцінюються можливості комерціалізації розробленого мобільного додатку та виведення на ринок IPIN (Indoor Navigation and Indoor Positioning).

1 ЗАДАЧА ВИЗНАЧЕННЯ ПОЗИЦІЇ ПРИСТРОЇВ У ПРИМІЩЕННЯХ

Для розгляду задачі позиціонування необхідне розуміння поняття навігації.

1.1 Задача навігації

Слово «навігація» має латинські корені та утворено зі слів «navis», що значить корабель та «agere» — лат. вести, рухати, направляти. Визначення навігації таке ж неоднозначне як і значення. Наприклад, Зоненберг та Хофман-Веленгоф визначили навігацію пов'язаною з об'єктами та машинами, але не людьми, а Прасад та Ругієрі зазначили, що люди можуть бути частиною процесу навігації [2, 3, 4]. Проте, деякі моменти є спільними для всіх визначень: зазначення початкової позиції або точки відправлення об'єкта чи людини; наявність точки призначення та інформації, необхідної, щоб знайти найкращий шлях з початкової в кінцеву. Ці три пункти є основними визначальними чинниками навігації, які необхідно враховувати до початку орієнтації.

У даній роботі обрано наступне визначення: процес успішної орієнтації людиною, машиною чи іншим об'єктом з точки відправлення через заздалегідь визначений та добре запланований шлях до бажаної точки прибуття в будь-якому середовищі.

У загальному, навігація — діяльність, що присутня з самого початку існування людей. Спочатку, термін «навігація» використовували, щоб описати рух кораблів в морі. В сучасному розумінні, він включає в себе рух людей, машин, роботів, літаків, космічних кораблів тощо.

Прогрес в морській та наземній навігації привів дослідників до роботи над процесом орієнтування в космосі. Ці ж дослідники спробували різноманітні підходи, щоб визначити позицію на Землі використовуючи орбітальні супутники. Успіхи експериментів розвинулись в розробку GPS (англ. Global Positioning System — глобальна система позиціонування). Саме GPS спростив навігацію та перетворив її в звичну діяльність. Система GPS широко застосовується зовні завдяки її точності та

ефективності. Однак всередині приміщень такого прийняття немає через те, що стіни та перекриття служать перешкодами для сигналів з супутників.

Услід за надзвичайним успіхом GPS, фокус досліджень був зміщений в сторону надання подібних сервісів для середовища всередині будівель [1]. Навігація в приміщеннях є темою для досліджень із застосуванням багатьох технік та технологій. Гу, Ло та Німегірс стверджують, що система навігації в будівлях складається з мережі пристроїв для знаходження об'єктів чи людей всередині будівель [5]. Ранні системи для відстеження та позиціонування включали в себе сенсори, що склались з передавачів та приймачів. Одна з частин (приймачі або передавачі) повинна бути зафіксована, в той час як інша є рухливою. Останні системи внутрішньої навігації засновані на смартфонах як рухливих пристроях.

1.2 Призначення програмного продукту

Метою магістерської дипломної роботи є розробка ефективної та легкої в налаштуванні підсистеми визначення позиції для мобільного додатку навігації з використанням доповненої реальності.

Це включає в себе наступні задачі:

- проаналізувати методи визначення позиції в приміщеннях;
- розробити сервіс внутрішнього позиціонування на основі вибраного підходу;
- інтегрувати підсистему з іншими складовими частинами програмного продукту.

Об'єктом дослідження є засоби визначення позиції в приміщеннях на основі комп'ютерного зору.

Предмет дослідження — візуально-інерційна система визначення позиції для потреб внутрішньої навігації в смартфонах.

У якості методу системи позиціонування був обраний оптичний. Обґрунтування вибору наведено в наступних розділах дисертації.

Комплексний програмний продукт — мобільний додаток для сучасних смартфонів, який дозволить здійснювати процес навігації в приміщеннях, тобто допоможе знайти точки інтересу та приведе до них в інтерактивному режимі.

Мобільний додаток повинен:

- визначати поточну позицію пристрою;
- надавати можливість перегляду схеми приміщення;
- забезпечувати можливість задання кінцевої точки навігації;
- підтримувати анотацію предметів та кімнат;
- виводити інструкції щодо переміщення в напрямку призначення;
- дозволяти замінити схему на іншу.

Враховуючи вищевказане, можна навести діаграму прецедентів для мобільного додатку (рисунок 1.1).

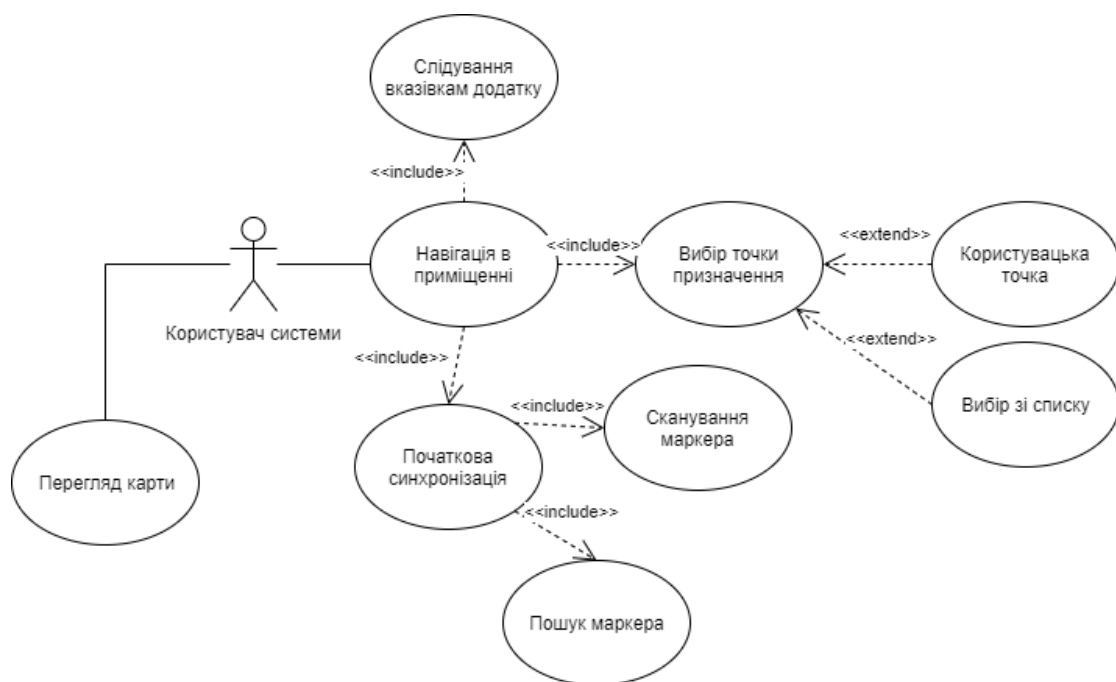


Рисунок 1.1 — Діаграма прецедентів комплексного продукту

Основним користувацьким сценарієм є навігація в приміщенні, що включає в себе послідовно початкову синхронізацію, вибір точки призначення та слідування до обраної точки, користуючись інструкціями додатку.

Підсистема, що розроблюється повинна:

- розпізнавати ознаки зображень для подальшої обробки;
- відстежувати зміни в положенні окремих ознак між кадрами;
- будувати хмару розпізнаних точок середовища;
- використовувати додаткові сенсори мобільних пристроїв для покращення точності;
- підтримувати відновлення позиції в разі втрати відслідковування;
- перетворювати позицію в локальній системі координат в глобальну, яка прив'язана до схеми будівлі з використанням маркерів;
- не потребувати встановлення дорогої інфраструктури, а лише розміщення друкованих маркерів.

Важливим є виділення основних задач програмного продукту та визначення, в яких цілях він буде використовуватися користувачами.

Потенційними сферами застосування є:

— сервіси на основі позиції. Тісно пов'язані з комерцією застосування, що використовують географічну позицію для доставки контекстно-залежної інформації, доступної на мобільному пристрої. Прикладами можуть служити отримання інформації про покази в кінотеатрах, концертах та інших подіях. Також до цієї категорії можна включити навігацію до потрібного магазину в торговому центрі або до офісу в загальнодоступній будівлі. На складі такі сервіси можуть бути цікавими як власнику так і його клієнтам. Особливо високим комерційним значенням характеризуються адаптивна реклама, локальний пошук послуг. На автобусних станціях та вокзалах системи можуть привести до потяга чи потрібної зупинки. Іншими прикладами є сповіщення про досягнення певної відстані до об'єкту, автоматичні процедури авторизації в компаніях.

— Медична турбота. В лікарнях інформація про позицію медичного персоналу в критичних ситуаціях є надзвичайно важливою. Іншим застосуванням в галузі медицини є відстеження пацієнтів та обладнання, наприклад, розпізнавання падіння людей. Також точне позиціювання необхідно під час операцій із застосуванням роботів.

— Розумний транспорт. Масовим застосуванням може бути проведення власників до їх автомобілів в підземних паркінгах.

— Промисловість. Для багатьох промислових застосувань знання про позицію в приміщенні є основним елементом функціонування, наприклад, для роботів, розумних фабрик, пристроїв автоматичного контролю якості та моніторингу. СВП можуть допомагати знаходити помічені інструменти та обладнання. Автоматичні системи безпеки, захисту працівників та уникнення зіткнень також засновані на застосуванні СВП.

— Музеї. Дослідження активності відвідувачів, визначення, що їм цікаво, а що не користується попитом. Також в музеях можна показувати контекстну мультимедіа інформацію поблизу експонатів.

— Логістика та оптимізація. Для потреб оптимізації процесу в складних системах базовою є інформація про розташування предметів та працівників. Наприклад, на великому складі важливо швидко знаходити потрібні товари, що стає можливим завдяки СВП.

— Моніторинг стану будівель. Датчики, вбудовані в сталі конструкції можуть надавати високоточні вимірювання зсуву, що виникають через високі навантаження та псування матеріалів. Зазвичай такі системи створюються на основі пасивних пристроїв радіочастотної ідентифікації.

Першочерговою ціллю є спрощення пошуку потрібних речей та приміщень в будівлі відвідувачами. Іншим корисним застосуванням в комерційних сферах може бути анонімний збір телеметрії для потреб маркетингу та статистики.

1.3 Вхідні дані

Вхідними даними є потік зображень з камери смартфона, покази вбудованих датчиків: акселерометр, гіроскоп та інші, а також схема поточного приміщення.

Перші два джерела є динамічними та зчитуються безперервно під час роботи програми за допомогою спеціальних інтерфейсів, що надаються операційною системою мобільного пристрою.

Схема приміщення повинна бути створена попередньо, до запуску програми. Вона потрібна для прив'язки локальної системи координат пристрою до глобальної.

1.4 Компоненти системи

У запропонованій системі можна виділити наступні складові-підсистеми:

- підсистема позиціонування, що розглядається;
- підсистема знаходження шляху;
- підсистема інтеграції моделі будівлі.

Усі компоненти є частиною одного мобільного додатку, тому взаємодія між ними є прямою та не покладається на зовнішні ресурси та протоколи, як то HTTP. Взаємодію між ними демонструє рисунок 1.2.

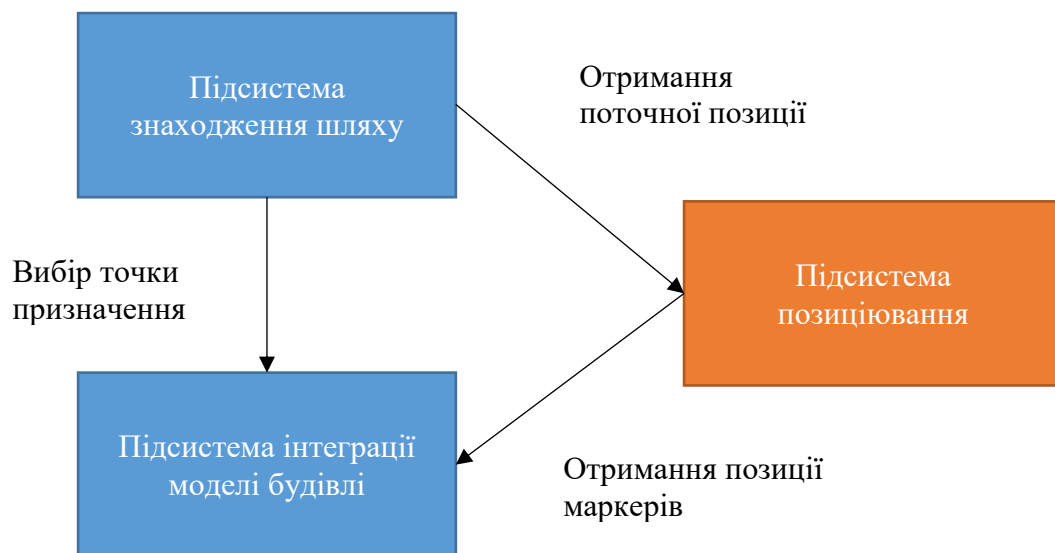


Рисунок 1.2 — Взаємозв'язок підсистем програмного продукту

Помаранчевим кольором виділена підсистема, що розробляється в цій роботі.

Висновки до розділу

Таким чином, в першому розділі були розглянуті передумови та мотивація створення програмного забезпечення для спрощення та оптимізації процесу навігації в приміщеннях, розглянута компонентна структура розробленої системи, поставлені перед нею задачі.

2 АНАЛІЗ ПРОБЛЕМИ ВИЗНАЧЕННЯ ПОЗИЦІЇ МОБІЛЬНОГО ПРИСТРОЮ

У розділі розглядаються теоретичні основи визначення позиції пристроїв, принципи, на яких можуть базуватися рішення задачі позиціонування мобільних телефонів.

2.1 Система внутрішнього позиціонування

Система внутрішнього позиціонування (IPS, СВП) — система, що включає мережу пов'язаних пристроїв для бездротового визначення позиції об'єктів та людей всередині будівель та частково закритих зонах [6].

2.1.1 Особливості внутрішнього позиціонування

Часто виникає питання: «Чому розділяють системи внутрішньої та зовнішньої навігації?». Насправді, принаймні теоретично, більшість систем позиціонування можуть використовуватись як зовні, так і в приміщеннях. Проте ці два оточення мають значні відмінності, що спричиняють різну якість роботи. Внутрішні середовища є складними для визначення позиції за декількома причинами:

- значне подовження шляху сигналу через відбиття від стін та фурнітури;
- умови непрямой видимості;
- високий ступінь атенюації (затінення) та розсіювання сигналу через вищу щільність перешкод;
- швидкі просторові зміни через присутність людей та відкриття дверей;
- необхідність високої точності.

З іншого боку, існують і деякі спрощення:

- малі площі покриття;
- незначний вплив погодних умов, малий градієнт температури та повільна циркуляція повітря;

- фіксовані геометричні обмеження у зв'язку з ортогональністю стін та наявності поверхонь, наближених до площин;
- доступність інфраструктури, такої як електромережа, доступ до мережі Інтернет;
- нижча динаміка процесу через повільну ходьбу.

2.1.2 Основні принципи визначення позиції

У системах навігації в будівлях під техніками позиціювання розуміють методи визначення та оцінки положення вузлів. Існує ряд алгоритмів отримання інформації про близькість та відстань, що засновані на вимірюванні сигналів та їх властивостей. Такі алгоритми перетворюють деякі властивості сигналів у відстані та кути і потім обчислюють фактичну позицію цільового об'єкта. Таким чином, користувач може отримувати інформацію про положення в процесі навігації [7].

Більшість методів, алгоритмів та складових технологій позиціонування не є новими, оскільки реалізуються також зовні. Однак те, як вони ведуть себе в приміщенні, відрізняється докорінно.

Щоб визначити позицію користувача, використовують дві складові: властивості сигналу та алгоритми позиціонування. Властивості сигналів — геометричні параметри, що складаються з таких метрик як кут, відстань та сила сигналу для отримання позиції об'єкта шляхом математичних обчислень. Існують різні методи вимірювання сигналів, але основними з них є кут прибуття (AOA), час прибуття (TOA), різниця в часі прибуття (DTOA) та індикація сили отриманого сигналу (RSSI).

У методі AOA кут і відстань обчислюються відносно двох або більше опорних точок через перетин напрямних ліній. Ілюстрація процесу зображена на рисунку 2.1.

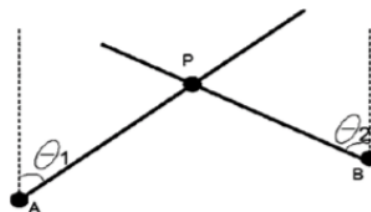


Рисунок 2.1 — Метод кута прибуття

Обчислення кута та відстані використовується для оцінки та визначення позиції передавача та ця інформація може бути застосованою для потреб позиціонування та навігації. На практиці, апаратне забезпечення зазвичай є складним та дорогим.

У той час як вимірювання АОА базується на кутах, ТОА (англ. Time of Arrival — час прибуття) засновується на відстанях. ТОА ще називають TOF (англ. Time of Flight — час польоту) тому, що відстежується час, який необхідний сигналу, щоб прийти до приймача від фіксованого передатчика, який водночас є опорною точкою. З іншої сторони TOF — час передачі сигналу від рухливого передатчика до фіксованого приймача. Час є однаковим, тому ці методи синонімічні. Різниця лише в тому, що вважати орієнтиром.

ТОА використовує абсолютний час прибуття, а не різницю часу між відправленням та прийманням. Метод надає високу точність, але за рахунок складності пристроїв. Метод TDOA (англ. Time Difference of Arrival — різниці часу прибуття) призначений для того, щоб прибрати таку складність апаратної частини. Рисунок 2.2 демонструє суть методу ТОА.

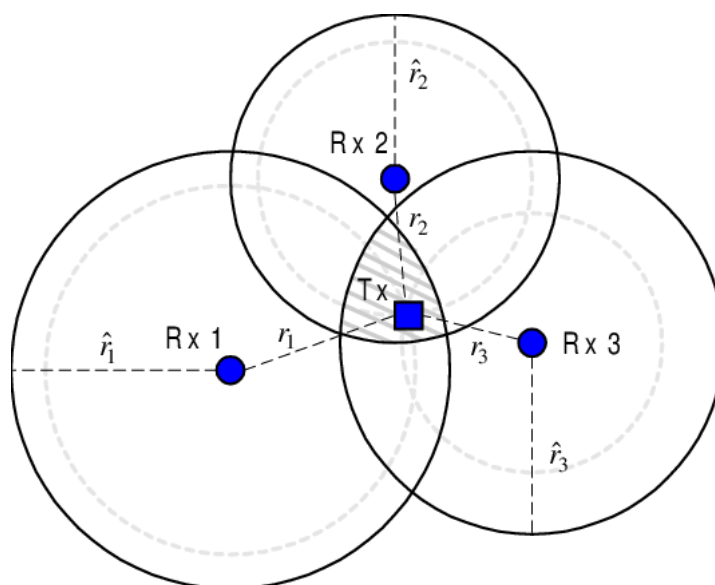


Рисунок 2.2 — Метод часу прибуття

Метод TDOA, так само як і ТОА, засновується на відстані. Він визначає відносну позицію мобільного передатчика на основі різниці часу поширення сигналу до декількох орієнтирних точок або сенсорів. Іншими словами, TDOA вимірює різницю між ТОА двох і більше сенсорів. Цим самим, він відкидає необхідність знати

час, коли сигнал був відправлений. TDOA теж надає високу точність. На рисунку 2.3 зображена ілюстрація методу різниці часу прибуття.

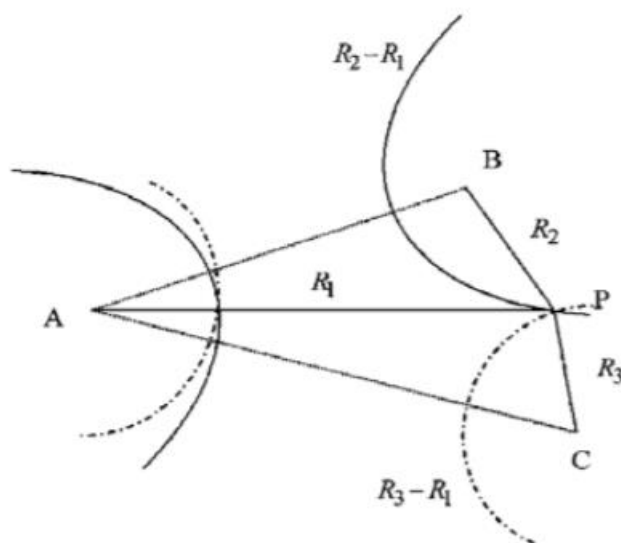


Рисунок 2.3 — Метод різниці часу прибуття

На відміну від метрик відстані та кутів, RSSI (англ. Received Signal Strength Indication — індикація сили отриманого сигналу) вимірює рівень потужності отриманого сигналу, присутнього в радіоінфраструктурі, який може бути використаний для оцінки відстані між мобільними пристроями. Іншими словами, підхід RSSI заснований на атенюації сигналів, обчислює зменшення сили сигналу через втрати на його шляху. Чим вищий RSSI, тим краща якість сигналу та ближче передавач. Проте, в приміщеннях важко підтримувати пряму лінію видимості, тому на позиціонування на основі RSSI може впливати затінення та відбиття, тим самим зменшуючи точність.

Підведемо підсумки, порівнявши описані властивості сигналів в таблиці 2.1.

Таблиця 2.1. Порівняння властивостей сигналів

Властивість сигналу	Метрика вимірювання	Переваги	Недоліки
Кут прибуття (АОА)	Кут	Точність на рівні кімнати	Складність, дороговизна, низька точність на великому покритті

Таблиця 2.1 (продовження)

Властивість сигналу	Метрика вимірювання	Переваги	Недоліки
Час прибуття (TOA)	Відстань	Висока точність	Складність, ціна
Різниця часу прибуття (TDOA)			Ціна
Індикація сили отриманого сигналу (RSSI)	Потужність сигналу	Низка вартість	Середня точність

Отже, властивості сигналів — важливий елемент для визначення позиції, який є обов’язковим в подальших обчисленнях. Вибір основної властивості впливає на кінцеві характеристики системи позиціонування, тому до нього необхідно ставитись відповідально.

Алгоритми позиціонування визначають як обрахувати позицію цільового об’єкта. Іншими словами, ці алгоритми транслюють вказані властивості сигналу у відстані та кути та обчислюють на основі них положення. Наприклад, коли відстані між об’єктом та опорними точками вже оцінені, алгоритм обчислює позицію.

Точність інформації про позицію залежить від правильності значення вимірюваної властивості сигналу. Алгоритми позиціонування мають унікальні переваги та недоліки, отже, одночасне використання більше одного типу алгоритмів позиціонування підвищує точність позиціонування та продуктивність. Тому існують багато прийомів для визначення положення за допомогою інформації про напрямок, відстань чи близькість на основі вимірювання сигналу. Однак основними алгоритмічними прийомами при позиціонуванні є тріангуляція, трилатерація, близькості та аналіз сцени. У відповідних алгоритмах позиціонування застосовуються різні властивості сигналу.

Тріангуляція (або ангуляція) використовує геометричні властивості трикутників для оцінки позиції об’єкту шляхом обчислення кутових вимірювань відносно двох відомих орієнтирних точок. Інакше кажучи, позиція об’єкту

знаходиться перетином двох пар напрямних ліній. АОА застосовується для обчислення відстані між напрямними лініями або фіксованими точками.

У трилатерації позиція знаходиться на основі вимірювань відстаней до трьох відомих точок. Мультилатерація працює аналогічно, але отримує інформацію з чотирьох та більше точок. Саме такий підхід використано для глобальної системи позиціонування GPS.

На відміну від тріангуляції та трилатерації, алгоритм близькості не дає абсолютної або відносної оцінки позиції, а лише деяку інформацію про позицію. Як приклад, розглянемо мережу антен з відомими заздалегідь позиціями. Мобільний пристрій в русі обирає одну найближчу антену з найсильнішим сигналом. Позиція пристрою визначається через RSSI, який оцінює відстань між антеною та смартфоном.

У алгоритмах оцінки сцени результат не залежить від кутів та відстаней. Аналіз сцени збирає інформацію про ознаки на ній та порівнює її з однією або декількома базами даних. Зібрану інформацію також називають «відбитком» (англ. fingerprint), тому що вона унікально характеризує конкретну зону на сцені та має відрізнитись від інших. Метод «відбитків» використовує раніше записану базу даних сили сигналу в зоні покриття. В результаті отримуємо кращу точність в порівнянні з методами оцінки поширення сигналів. Проте, фаза збору даних потребує велику кількість часу та зусиль для побудови карт сили сигналу для кожної точки доступу Wi-Fi. В результаті, метод відбитків є дорогим, складним та повільним.

Таблиця 2.2 підводить підсумки по алгоритмам позиціонування.

Таблиця 2.2. Порівняння алгоритмів позиціонування

Алгоритм	Властивість сигналу	Переваги	Недоліки
Тріангуляція	АОА	Простий, дешевий, точний в межах однієї кімнати	На великій зоні покриття стає дорогим та складним
Трилатерація	ТОА, TDOA	Точний	Складний, дорогий

Таблиця 2.2 (продовження)

Алгоритм	Властивість сигналу	Переваги	Недоліки
Близькості	RSSI	Простий	Часто низька точність, дорогий
Аналіз сцени (відбитків)	RSSI	Не потребує додаткових пристроїв	Середня точність, може потребувати багато часу для створення карти покриття

Орієнтовна класифікація технік позиціонування наведена на рисунку 2.4.



Рисунок 2.4 — Основні техніки позиціювання

Визначення поточної позиції користувача — найбільш важлива, водночас, найважча фаза внутрішньої навігації. Розглянемо, проаналізуємо та порівняємо технології, що застосовуються для позиціонування.

2.1.3 Технології позиціонування

Почнемо з хвиль інфрачервоного спектру, їх переваг та обмежень. Однією з ранніх систем навігації була Active Badge (1992 рік) [8]. Система активних знаків є

упізнаваною системою позиціонування об'єктів та людей, що носять спеціальну мітку. Система складається з мережі інфрачервоних сенсорів, пов'язаних дротами та підключених до центрального сервера. Хоча сенсори є дешевими, дослідники погоджуються, що в цілому така система є дороговартісною. В загальному, система на базі інфрачервоних міток має довгий період до заміни батареї. Активні мітки застосовують ТОА та трилатерацію для вказання положення.

У подальшому бажання розробити систему без міток та бейджів мотивувало дослідників запропонувати пасивну інфрачервону систему позиціонування з використанням пасивних термальних сенсорів, відомих як термостовпчики. Такі датчики вимірюють теплове випромінювання людей в радіусі їх дії. Це є можливим, бо температура людського тіла відрізняється від температури в кімнаті. Проте, люди — не єдине джерело тепла, інші нагрівачі здійснюють негативний вплив на процес, наприклад, лампи розжарювання, електронні пристрої тощо.

Серед інших недоліків IR — дуже обмежений радіус дії та точність. Для великих зон потрібно багато сенсорів, що призводить до складності та дороговизни інфраструктури в цілому. Також інфрачервоні хвилі інтерферують з денним світлом. Тому дана технологія не набула подальшого розвитку та фокус досліджень зсунувся в сторону більш перспективних підходів, таких як ультразвук.

Ультразвукові системи позиціонування показують високу точність на рівні однієї кімнати. Вони включають ультразвукові вузли, які служать як приймачі або передавачі. В якості прикладів можна навести системи Active Bat, Cricket, Dolphin [9].

Такі властивості ультразвукового сигналу як низька швидкість поширення в порівнянні з електромагнітними хвилями, незначне проникнення через стіни та низька вартість обладнання роблять такий підхід цікавим з точки зору досліджень. З іншого боку, імплементація на великих областях нівелює переваги таких систем; вони страждають від шуму, відбиття та інтерференції.

На відміну від ультразвуку, який має обмежену дальність поширення через високий ступінь затінення під час проходження в повітрі, системи визначення позиції чутного звуку не є лімітованими в цьому аспекті. В них застосовуються звукові плати та динаміки повсякденних пристроїв.

Такі системи, зазвичай, складаються з мобільних телефонів, акустичних пристроїв, центрального серверу та бездротової мережі. Смартфон передає звук на приймачі. Коли відбувається запит та визначення позиції з телефону, динамік пристрою випромінює звуковий сигнал, який сприймається ресіверами. Кожний з приймачів оцінює час прибуття сигналу та надсилає його на центральний сервер разом з інформацією про власне розміщення в просторі через мережу. Центральний сервер на основі цих даних обчислює ймовірне положення смартфона та відправляє назад. Мобільний телефон може також виступати приймачем, тоді ролі інвертуються.

Значними недоліками є низька частота оновлення, обмежений діапазон частот мікрофонів. Також виникають питання щодо приватності даних користувачів.

Наступною технологією на розгляд є магнітне позиціонування. Вона є однією з найстаріших — перші дослідження були опубліковані в 1979 році і все ще залишаються актуальними [10]. Магнітні системи позиціонування залучають використання магнітного поля. Об'єкти з феромагнітних сплавів впливають на магнітне поле поряд з ними, так само як і електричні пристрої та проводи.

Можна також згенерувати статичне магнітне поле, розмістивши постійні магніти та котушки, по яких пускають змінний або пульсуючий постійний струм. Маутц класифікував дослідження за такими категоріями [1]:

- магнітне поле зі струмів;
- використання постійних магнітів;
- зняття «магнітних відбитків».

Таким чином, магнітні СВП не страждають від ефектів непрямої лінії видимості. Недоліки — вплив структур будівлі на поле, потрібна велика кількість магнітних датчиків.

Радіочастотна СВП використовує сигнали радіочастот та відповідну інфраструктуру. Такі системи мають перевагу в тому, що аналоговий сигнал може проникати через перешкоди та стіни, що призводить до перевикористання існуючих пристроїв, які випромінюють такий сигнал та більшої зони покриття. Радіочастотні СВП надалі поділені на підкатегорії: Bluetooth, Ultra-wideband (UWB), Wireless Sensor Network (WSN), Wireless Local Area Network (WLAN), Radio-Frequency Identification

(RFID) та Near Field Communication (NFC). Кожна з них має унікальні переваги та недоліки.

Зупинимось лише на Bluetooth та WLAN як на найпопулярніших радіочастотних системах ВП.

Bluetooth — безпроводна технологія для обміну даними на короткій відстані. СВП Bluetooth складається з мобільного пристрою з підтримкою Bluetooth, маяків та серверу, приєднаних до мережі. Радіус дії технології складає від декількох до 100 метрів в залежності від потужності (класу) випромінювача. Пристрої в зоні дії можуть пов'язуватись один з одним для передачі даних. Переваги від застосування Bluetooth для внутрішньої навігації полягають в малому енергоспоживанні, малому розмірі, невисокій вартості маяків. З іншого боку, процес знаходження маяків є повільним за своєю суттю, впливаючи тим самим на навігацію в реальному часі [11].

Аналогічно до раніше описаного рішення, WLAN-базована система використовує високочастотні хвилі для визначення положення в зоні їх покриття. Відмінністю є те, що використовуються звичні точки доступу Wi-Fi замість Bluetooth маяків. Основною технікою є метод «відбитків», в якому збирається карта відповідності сили сигналу до позиції в кімнаті.

Рисунок 2.5 надає перелік найбільш популярних та широко застосовуваних технологій систем позиціонування.

Технології позиціонування	Інфрачервоні хвилі
	Ультразвук
	Звук чутного спектру
	Магнітне поле
	Комп'ютерний зір
	Хвилі видимого спектру
	Радіохвилі
	Навігаційне обчислення
	Гібридні

Рисунок 2.5 — Технології позиціонування

Оптична система позиціювання — така система, в якій положення об'єкта визначається шляхом розпізнавання зображення з допомогою мобільного сенсора чи камери мобільного пристрою, що переноситься користувачем [12].

У методі на основі маркерів камера мобільного телефону отримує візуальну інформацію, використовуючи маркери, наприклад QR-код. Система складається з мобільного пристрою з камерою, QR-коду та сервера [13]. Камера мобільного пристрою використовується для зйомки даних шляхом сканування QR-коду, в той час як сервер використовується для цілей відстеження та зберігання інформації, наприклад, схеми поверху для потреб пошуку.

Останніми роками смартфони стали наскільки потужними, що з'явилися підходи без використання маркерів. Відстеження переміщення телефону відбувається завдяки розпізнаванню ознак зображення та узгодження їх між кадрами.

Проблема позиціювання на основі зображення була і є актуальною в останні десятиліття. У 1998 році група технологій бачення обговорила майбутнє життя людей. І однією з основних технологій була локалізація людей в будинку, використовуючи відеоспостереження. Методи позиціювання, базовані на баченні, можна поділити на 3 категорії. Перша категорія використовує моделі будівель. Такі методи розпізнають об'єкти на зображенні та перевіряють співпадіння ознак з бази даних будівлі. Хіль та Боріелло порівнювали фото з планом поверху [14]. Кохоутек та інші виявляли конкретні об'єкти в точкових хмарах та порівнювали їх з моделлю інтер'єру CityGML для визначення пози смартфона [15]. Друга категорія методів візуального позиціонування заснована на зображеннях. Такі підходи в основному порівнюють схожість між тестовим зображенням та еталоном, що був знятий раніше в офлайн-фазі та виводять місце розташування еталоного зображення з найвищою оцінкою схожості.

Кім і Джун порівнювали поточний вид із камери з послідовностями зображень, що зберігаються в базі даних [16]. Їх метод розроблений для програм доповненої реальності з функцією отримання додаткової інформації. Вернер та інші оцінювали положення та орієнтацію за допомогою опорних зображень та інформації про місцезнаходження, отриманої від побудованої заздалегідь бази даних [17]. Вони

розробили алгоритм для оцінки відстані від точки огляду до зображення.

Остання категорія використовує розгорнуті закодовані цілі, включаючи концентричні кільця, штрих-коди або візерунки з кольорових крапок тощо. Мулонні та інші розміщували штрих-коди у різних місцях так, щоб камери могли захопити ці позначки та отримати місцезнаходження, а також іншу інформацію [18].

У загальному, оптичні системи позиціонування використовують камеру та обчислювальні потужності мобільного пристрою. Сучасні смартфони поставляються з такими інерційними датчиками як акселерометри, гіроскопи та магнітометри. Звідси, маємо значне зменшення компонентів інфраструктури, які потребують встановлення. До того ж, система коштує значно менше в порівнянні з іншими СП. Проте, можна відзначити такі недоліки як відносно низька точність, негативний вплив багатьох ефектів як занадто яскраве світло та розмиття під час руху, значні накопичувальні похибки. Також існує ризик, що середовище зміниться і попередньо записані схеми та хмари точок перестануть бути правильними [19].

Популярним методом в сучасних оптичних СВП є SLAM. Розглянемо суть його роботи та складові.

2.2 Метод одночасної локалізації та картографування SLAM

Метод SLAM (англ. Simultaneous Localization and Mapping) — популярна техніка комп'ютерного бачення, що прийшла до нас із спільноти робототехніки для автономних транспортних засобів. Процес відображення SLAM намагається отримати просторові дані (тривимірні хмари точок) із середовища для того, щоб побудувати глобальну карту. Водночас, він відстежує позицію суб'єкта.

Алгоритми SLAM обмежуються наявними ресурсами, таким чином не можуть бути абсолютно досконалими, бо досягають оперативної доступності. Опубліковані методи і підходи реалізовані в безпілотних автомобілях, безпілотних літаючих засобах, автономних підводних апаратах, планетоходах, згодом виникли в побутових роботах і навіть всередині людського тіла [20].

2.2.1 Розвиток SLAM

Вирізняють три ери розвитку технологій SLAM. Класичну еру (1986-2004 роки) характеризують створенням вірогіднісних формулювань SLAM, включаючи підходи на основі розширених фільтрів Калмана, фільтрів часток Рао-Блеквела та оцінки максимальної схожості [21]. Також в цей час були окреслені базові складнощі, пов'язані з ефективністю та надійністю асоціації даних. Другим періодом є ера алгоритмічного аналізу (2004-2015). Впродовж неї ми побачили дослідження фундаментальних властивостей SLAM, включаючи спостережність, збіжність та послідовність, були розроблені основні реалізації методу з відкритим кодом [22]. Наразі ми входимо до третьої ери — стійкого сприйняття, що характеризується наступними вимогами:

- Надійність: система SLAM функціонує з низькою частотою помилок протягом довгого періоду часу в широкому наборі оточень; включає в себе механізми запобігання збоїв та має можливості автоматичного налаштування;

- Високий ступінь розуміння: система йде далі, ніж реконструкція примітивних геометричних конструктів на шляху до високорівневого розуміння оточення (семантика, фізика, доступність);

- Обізнаність ресурсів: SLAM пристосовується до наявних обчислювальних та сенсорних ресурсів;

- Задаче-орієнтоване сприйняття: система здатна обирати релевантну інформацію та фільтрувати зайві сенсорні дані, надає адаптивні репрезентації, складність яких може змінюватись в залежності від конкретної задачі.

Різноманітні алгоритми SLAM реалізовані у відкритих бібліотеках Robot Operating System, які часто використовуються з Point Cloud Library для побудови 3D карт або візуальних ефектів у OpenCV. Список відомих реалізацій: LSD SLAM, ORB SLAM2 [23, 24].

Обираючи як інтегрувати виміряні ознаки з різних зображень: оцінюючи розподіл ймовірностей або оптимізуючи, проблему оцінки можна вирішити підходами на основі фільтрів або коригування набору (англ. bundle adjustment).

MonoSLAM вважається першим підходом з використанням фільтрів щодо переносу проблеми SLAM з спільноти робототехніки на інші галузі [25]. Робота Лонге-Хігінса призвела до появи багатьох технік SfM [26]. Він дослідив, що обчислення відносної позиції є можливим шляхом пошуку 8 точок-відповідностей між двома кадрами в епіполлярній геометрії. Крім того, з'явилися інші точки зору на вирішення проблем факторизації та усереднення повороту.

Архітектура системи SLAM включає в себе два основних компоненти: так звані зовнішня та задня частини (front-end та back-end відповідно). Архітектура та взаємодія компонентів узагальнена на рисунку 2.6.

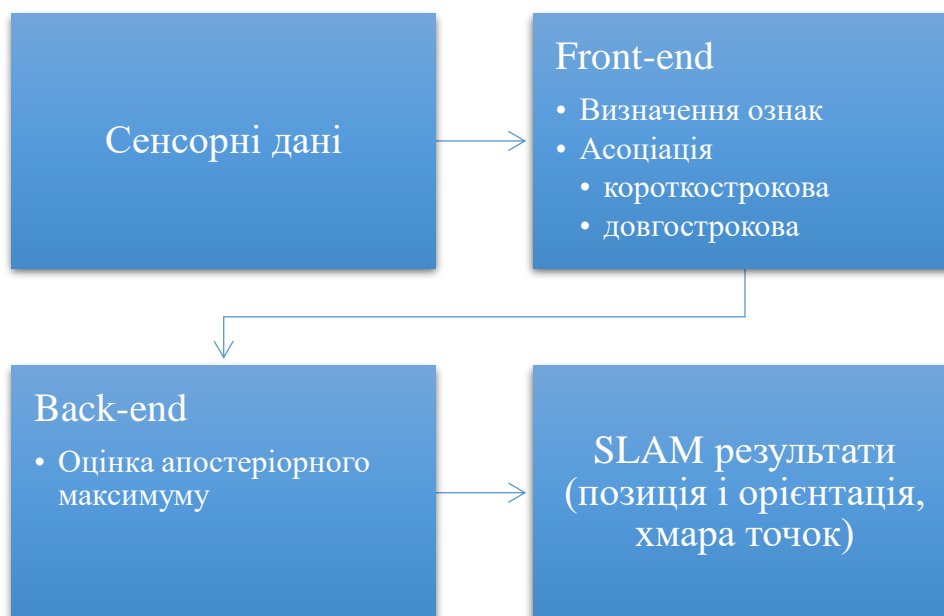


Рисунок 2.6 — Процес обробки даних в SLAM

Зовнішня частина абстрагує сенсорні дані в моделі, що підходять для оцінки. Задня частина виконує виведення нових даних на основі раніше побудованих моделей.

2.2.2 Постановка задачі SLAM

Поточним де-факто стандартним формулюванням SLAM є проблема оцінки апостеріорного максимуму [27]. Часто в формулюваннях використовують графи факторизації для вираження взаємозалежностей між змінними.

Припустимо, що ми хочемо оцінити невідому змінну X . В SLAM X зазвичай означає траєкторію об'єкту (дискретна множина поз) та позиції орієнтирів в оточенні. Ми отримуємо множину вимірювань $Z = \{z_k: k = 1 \dots m\}$, таких, що кожне вимірювання може бути виражене як функція від X , тобто $z_k = h_k(X_k) + e_k$, де $X_k \subseteq X$ є підмножиною змінних, h_k — відома функція (модель вимірювання) і e_k — випадковий шум вимірювання.

В оцінці апостеріорного максимуму ми оцінюємо X шляхом обчислення змінних X^* , при яких набуває максимум апостеріорна ймовірність $p(X|Z)$:

$$X^* = \underset{X}{\operatorname{argmax}} p(X|Z) = \underset{X}{\operatorname{argmax}} p(Z|X)p(X), \quad (2.1)$$

де $p(Z|X)$ — вірогідність вимірювань Z , при заданих значення X ;

$p(X)$ — попередня ймовірність над X .

В формулі (2.1) рівність слідує з теореми Байеса. Попередня ймовірність включає будь-які існуючі знання про X . Якщо попередніх знань немає, $p(X)$ стає постійною (рівномірний розподіл), що не впливає на процес і може бути викинутою з оптимізації. У цьому випадку оцінка МАЙ зводиться до оцінки максимуму ймовірності. Зауважимо, що на відміну від фільтрів Калмана, оцінка МАЙ не вимагає явної відмінності між моделлю руху та спостереження: обидві моделі розглядаються як фактори і безперебійно включені в процес оцінки. Більше того, варто зазначити, що фільтрація Калмана та оцінка МАЙ повертають однакову оцінку в випадку лінійного гаусіана, хоча в загального випадку це не так.

Вважаючи, що вимірювання Z є незалежними (тобто шуми не є пов'язаними), проблема (2.1) перетворюється в:

$$X^* = \underset{X}{\operatorname{argmax}} p(X) \prod_{k=1}^m p(z_k|X) = \underset{X}{\operatorname{argmax}} \prod_{k=1}^m p(z_k|X_k) \quad (2.2)$$

В правій частині z_k залежить тільки від підмножини змінних в X_k . Проблема (2.2) може бути інтерпретована в термінах виведення графу факторизації. Змінні

виступають вузлами в графі. Терми $p(z_k|X_k)$ та $p(X)$ називаються факторами та кодують ймовірнісні обмеження над підмножиною вузлів. Граф факторизації є графічною моделлю, що виражає залежність між k -тим фактором та відповідними змінними X_k .

Щоб записати (2.2) в більш явній формі, припустимо, що шум вимірювання e_k є гаусівським шумом з інформаційною матрицею Ω_k (обернена до матриці суміжності). Тоді формула (2.2) перетворюється в наступну форму:

$$p(z_k|X_k) \propto \exp\left(-\frac{1}{2} \|h_k(X_k) - z_k\|_{\Omega_k}^2\right), \quad (2.3)$$

де ми використовуємо нотацію $\|e\|_{\Omega}^2 = e^T \Omega e$. Аналогічно, припустимо, що попередня ймовірність може бути записана як $p(X) \propto \exp\left(-\frac{1}{2} \|h_0(X) - z_0\|_{\Omega_0}^2\right)$ для деякої заданої функції h_0 , середнього значення z_0 та інформаційної матриці Ω_0 . Так як максимізація апостеріорної ймовірності є тим же самим як мінімізація від'ємної логарифмічної апостеріорної ймовірності, формула (2.2) набуває вигляду (2.4):

$$X^* = \underset{X}{\operatorname{argmin}} -\log(p(X) \prod_{k=1}^m p(z_k|X_k)) \quad (2.4)$$

$$X^* = \underset{X}{\operatorname{argmin}} \sum_{k=0}^m \|h_k(X_k) - z_k\|_{\Omega_k}^2 \quad (2.5)$$

Постановка (2.5) є нелінійною проблемою найменших квадратів, як і багато інших задач в області робототехніки та комп'ютерного зору.

Можна помітити схожість постановки задачі SLAM (2.5) та коригування набору в підході Structure from Motion [28]. Дійсно, обидва процеси виходять з формулювання оцінки МАЙ. Проте, існує дві особливості, що роблять SLAM унікальним. По-перше, фактори в формулюванні не обмежуються лише моделлю геометричної проекції як в SfM, а можуть включати широкий спектр сенсорних моделей, наприклад, інерційних датчиків, GPS тощо. До прикладу, в картографії на основі лазерів фактори часто обмежують відносні пози щодо точок огляду, тоді як в

прямих методах візуального SLAM фактори «штрафують» зміни в інтенсивностях пікселів тієї ж частини сцени. Іншою відмінністю в порівнянні з SfM є те, що задача типово має вирішуватись інкрементально; нові вимірювання доступні через певний фіксований проміжок часу в процесі руху.

Задача мінімізації зазвичай вирішується послідовним приведенням до лінійного виду, наприклад, методами Гауса-Ньютона або Левенберга-Маркарда. Приведення є ітеративним, починаючи з початкової здогадки \hat{X} та апроксимує функцію витрат з квадратичною вартістю, яка може бути оптимізована шляхом розв'язування набору лінійних рівнянь (так звані нормальні рівняння).

Основне розуміння сучасних вирішувачів SLAM полягає в тому, що матриця, що з'являється в нормальних рівняннях, є розрідженою і її розріджена структура продиктована топологією графу факторизації. Це дає можливість використовувати швидкі лінійні розв'язувачі. Більше того, це дозволяє проектувати інкрементальні (або online) рішення, які оновлюють оцінку X в процесі того як отримуються нові спостереження. Сучасні бібліотеки SLAM (наприклад, GTSAM, g2o, Ceres, iSAM та SLAM++) здатні вирішувати задачі з десятками тисяч змінних за кілька секунд [29].

Оцінка МАЙ виявилася більш точною та ефективнішою, ніж оригінальні підходи до SLAM, засновані на нелінійних фільтрах. Зауважимо, що деякі системи SLAM на базі EKF також продемонстрували чудові результати. Відмінними прикладами таких систем на базі EKF є фільтр Калмана, модифікований Мурікісом та Румеліотисом та системи візуальної навігації Коттаса і Геша [30, 31].

Оцінка апостеріорної ймовірності зазвичай виконується на попередньо опрацьованій версії даних з сенсорів. Саме тому, цей процес часто називають задньою частиною SLAM.

2.2.3 Зовнішній модуль SLAM та його задачі

На практиці може бути проблематичним записати вимірювання сенсорів як аналітичну функції стану, що вимагається в оцінці МАЙ. Наприклад, якщо необробленими даними датчика є зображення, може бути важко виразити інтенсивність кожного пікселя як функцію стану SLAM; така ж складність виникає з

більш простими датчиками (наприклад, лазер з одним променем). В обох випадках питання пов'язане з тим, що ми не в змозі створити достатньо загальне, але простежуване представлення навколишнього середовища; навіть за наявності такої загальної репрезентації було б важко написати аналітичну функцію, що з'єднує вимірювання з параметрами такого представлення.

З цієї причини перед бекендом SLAM широко розповсюдженим є практика мати модуль, який називається переднім або зовнішнім (front-end), що створює відповідні ознаки з даних датчиків. Наприклад, у системі SLAM на основі бачення, така підсистема витягує піксельне розташування з декількох помітних точок в оточенні. Піксельні спостереження цих точок тепер легко моделювати в рамках бекенду. Передня частина також відповідає за зв'язок кожного вимірювання з конкретним орієнтиром (скажімо, 3D-точка) у навколишньому середовищі, так звану асоціацію даних. Більш абстрактно, модуль асоціації пов'язує кожне вимірювання z_k з підмножиною невідомих змінних X_k так, що $z_k = h_k(X_k) + e_k$. Нарешті, front-end може також забезпечити початкову здогадку для змінних в нелінійній оптимізації (2.5). Наприклад, у монокулярному SLAM на основі ознак, зазвичай, фронтенд відповідає за ініціалізацію орієнтирів, триангулюючи їх позицію із кількох точок зору.

Існує два типи асоціації: короткострокова та довгострокова. Перша з них відповідає за пов'язання відповідних ознак в послідовних вимірюваннях. Наприклад, вона відстежить той факт, що 2 пікселі в послідовних кадрах вказують на одну й ту ж точку в тривимірному просторі. З іншої сторони, довгострокова асоціація (або як її ще звуть закриття петлі — англ. loop closure) займається встановленням відповідностей з старішими орієнтирами.

Попередня обробка, яка відбувається в фронтенді є залежною від конфігурації датчиків, так як поняття ознаки варіюється в залежності від потоку вхідних даних, який ми розглядаємо.

2.2.4 Типи помилок та способи боротьби з ними

Система SLAM підвержена помилкам в багатьох аспектах: відмови можуть бути алгоритмічними, програмними або апаратними. Перший клас включає типи

відмов, викликані обмеженнями існуючих алгоритмів SLAM (наприклад, труднощі в роботі з надзвичайно динамічними або складними середовищами). Останній же включає помилки, причиною яких є неточності сенсорів. Протидія цим помилкам є надзвичайно важливою для забезпечення довгострокової роботи системи.

Одним з основних джерел алгоритмічних неточностей є асоціація даних. Як вказано раніше, модуль асоціації займається пов'язанням частини нового вимірювання з деякими попереднього стану. У візуальному SLAM він асоціює ознаку з орієнтиром. Феномен псевдонімів сприйняття, в якому різна сенсорна інформація має однакове модельне представлення, робить вирішення даної проблеми складним. В разі присутності такого явища, створюються помилкові асоціації «вимірювання-стан», що в свою чергу призводить до неправильних оцінок. З іншої сторони, коли модуль асоціації помилково вважає вимірювання неправильним і відкидає його, то в подальших кроках використовується менше сенсорних даних, що знижує точність.

Ситуація стає гірше з присутністю немодельованої динаміки в оточенні (короткочасні та сезонні зміни). Часто припускають, що все навколо є статичним та не змінюється під час роботи системи (іншими словами, орієнтири статичні). Таке припущення є правдивим тільки в разі короткострокової роботи при відсутності об'єктів та людей, що рухаються в кадрі.

Проблема некоректної асоціації може бути вирішена в фронтенді та/або в бекенді SLAM системи. Традиційно, таку відповідальність покладають на фронтенд. Легше вирішувати проблему з короткостроковою асоціацією: якщо частота оновлення сенсора є відносно великою, порівняно з динамікою переміщень несучого об'єкта, то відстежувати ознаки, що належать одному орієнтиру досить легко. Інтуїтивно, висока частота оновлення означає, що точка зору не змінюється радикально, тобто ознаки в момент часу $t + 1$ будуть схожими на ті, які спостерігались в момент t . Довгострокова асоціація є складнішою та включає визначення та перевірку закриття циклів. Тут підхід «грубої сили» (англ. brute-force) є непрактичним. Моделі на основі візуальних слів вирішують невіддатливість пошуку довгострокових пар шляхом поділу простору на кванти, щоб забезпечити більш ефективний пошук [32]. Вони можуть бути складені в ієрархічні словникові дерева.

Техніки на основі таких моделей показали надійність роботи в межах однієї сесії. Проте, ці підходи не здатні обробляти значні зміни в освітленні сцени, так як слова перестають бути схожими.

Валідація закриття циклу, в свою чергу, включає в себе додаткові кроки геометричної перевірки, щоб бути впевненим в якості процесу. У візуальних застосуваннях часто використовують підхід RANSAC [33].

Не зважаючи на прогрес в задачі покращення надійності асоціації, через явище псевдонімів сприйняття, неможливо повністю уникнути передачу неправильних даних в бекенд. Неправильне закриття циклу може сильно знизити якість оцінки МАЙ. Задля вирішення такої проблеми, останнім часом пропонують підходи, що роблять бекенд більш стійким проти неправдивих показань. Деякі з них ґрунтуються на перевірці залишкової помилки, що диктується обмеженнями в процесі оптимізації [34].

У динамічному оточенні ситуація є двоякою. По-перше, система SLAM має визначати, відхиляти або приймати зміни. В той час як популярні підходи намагаються ігнорувати динамічну частину оточення, деякі включають рухомі елементи в модель як складову її частину. По-друге, SLAM має моделювати постійні чи напівпостійні зміни та розуміти коли та як оновлювати карту. Сучасні реалізації або підтримують карти постійними протягом певного часу (залежність від часу) або ж мають єдину презентації параметризовану деяким параметром, залежним від часу.

Відкритими проблемами є: відновлення системи після збоїв, протидія апаратним збоям, релокалізація в різних умовах, пристосування до змін в оточенні, автоматичне налаштування параметрів процесів.

2.2.5 Моделі представлення оточення

Метричне представлення (метрична карта) — символічна структура, що кодує геометрію оточення. Геометричне моделювання в двовимірному просторі є простішим, включає 2 основні парадигми: орієнтирні карти та карти зайнятості. Перші моделюють оточення як розріджений набір орієнтирів, останні дискретизують простір в сітку та назначають ймовірність зайнятості кожного елементу.

Розуміння того, як ефективно змодельовати тривимірний простір для картографії ще тільки створюється.

Більшість методів SLAM представляють сцену як розріджену множину 3D-орієнтирів відповідно до характерних об'єктів в просторі (лінії, кути тощо). Часто припускають, що орієнтири вирізняються, тобто сенсорні дані не лише представляються деякі геометричні аспекти орієнтиру, але надають деякий (можливо неточний) спосіб поєднати орієнтир з даними [35].

На противагу орієнтирним підходам, щільні представлення намагаються надати модель високої роздільної здатності тривимірної геометрії. Вони більше підходять для задач уникнення перешкод, візуалізації та рендерингу. В них необроблені представлення описують геометрію як великий неструктурований набір точок (хмара точок) або полігонів (полігонний суп) [36]. Вони набули популярності в монокулярному SLAM в поєднанні з використанням прямих методів, що оцінюють траєкторію руху та модель прямо з інтенсивності всіх пікселів зображення.

Граничні та просторово-розділові щільні репрезентації йдуть далі неупорядкованих наборів низькорівневих примітивів (наприклад, точок) та явно представляють поверхні та об'єми. Вони показують себе найкраще в таких задачах як планування руху, уникнення перешкод, маніпуляції та інших, пов'язаних з фізичними об'єктами. Граничні представлення визначають 3D-об'єкти в поняттях меж їх поверхонь. Простим прикладом є моделі, що засновані на площинах. Більш загальними граничними представленнями є:

- на основі кривих — тензорний добуток NURBS або B-сплайнів;
- сітчасті моделі поверхні — набір пов'язаних полігонів;
- явне задання поверхні.

Останнє задає поверхню як перетин з нульовою площиною функції заданої в \mathbb{R}^3 . Прикладами функцій є радіально-базисні, відстаней зі знаком та її обмежені версії.

Підходи з використанням ознак є досить зрілими, з довгою історією успіху. Вони дозволяють будувати точні SLAM системи з автоматичною релокалізацією та закриттям циклів. Проте такі системи залежать від наявності характерних ознак в

оточенні, правильності встановлення параметрів та найчастіше вони оптимізовані для швидкості за рахунок точності. Прямі методи працюють з піксельною інформацією та отримують користь з усіх пікселів на зображенні, навіть з тих, де градієнти малі. Тому вони можуть побити будь-які ознакові методи в одноманітно текстурованих сценах. З недоліків, вони вимагають багато обчислювальних потужностей (потужні графічні процесори) для роботи в реальному часі. Напівщільні методи працюють тільки з тими пікселями, які вирізняються (наприклад, грані та лінії), тим самим зменшуючи вимоги до потужностей машини; напівпрямі методи одночасно використовують і ознаки і піксельні зони.

Прогнозують, що в майбутньому будуть домінувати високорівневі представлення, включаючи об'єкти та фігури. Вони дозволяють надавати об'єктам фізичні властивості такі як об'єм та масу, що корисно для роботів, які взаємодіють зі світом [37].

Підсумовуючи, процес складається з 4 кроків:

- Отримання даних з сенсорів: для мобільних пристроїв це зазвичай включає камеру, акселерометр та гіроскоп. Можуть бути доповнені іншими сенсорами, як то GPS, датчик світла, сенсор глибини тощо.

- Front-End: відповідає за розпізнавання ознак. Також пов'язаний з орієнтирами — ключовими тривимірними точками. Довгострокова асоціація зменшує похибку, розпізнаючи місця, які вже зустрічались раніше.

- Back-End: займається встановленням відношень між різними кадрами, визначає позу камери (позиція та орієнтація). Деякі алгоритми створюють розріджену реконструкцію на основі ключових точок. Інші ж намагаються зафіксувати щільну хмару 3D-точок навколишнього середовища.

- Оцінка: отримання результату, що містить ознаки, їхні позиції та позицію камери в середовищі.

Таким чином, SLAM та такі пов'язані техніки як візуально-інерційна одометрія все більше розгортаються у різноманітних реальних умовах, від безпілотних автомобілів до мобільних пристроїв. На технології SLAM будуть все більше покладатися в наданні надійного метричного позиціонування в ситуаціях, коли

інфраструктурні рішення, такі як GPS, недоступні або не забезпечують достатньої точності. Можна передбачити створення хмарних сервісів позиціонування, так як локалізація є важливою в сучасному світі.

2.3 Аналіз існуючих програмних систем

Ще зовсім нещодавно задачі визначення позиції стосувались виключно робототехніки. Алгоритми тестувались на роботах та використовувались в процесі їх переміщення. Наразі ж, абсолютно ясно, що люди можуть отримати багато переваг від портативних систем навігації. Саме тому активно розробляються нові програмні та апаратні засоби для вирішення даної проблеми.

Шляхом аналізу результатів пошуку в мережі за ключовими словами «indoor navigation» розглянемо існуючі програмні системи внутрішнього позиціонування.

2.3.1 Системи зі спеціальними пристроями

Першою в результатах пошуку Google є німецька компанія infSoft [38]. Вона працює з 2005 року в сфері надання позиційних сервісів. Найбільшими її клієнтами є аеропорт у місті Франкфурт, Siemens, Swiss Federal Railways та Roche. Вважається лідером ринку.

У якості технологій, що застосовуються, заявляються Wi-Fi, Bluetooth Low Energy, Ultra Wideband, RFID. Ними пропонується пристрій, що поєднує в собі всі вказані сенсори, тобто є універсальною апаратною одиницею розгортання мережі для внутрішньої навігації. Також існує програмна платформа infSoft LocAware (англ. Location Awareness). Вона містить в собі багато модулів, основними з яких є:

- засоби редагування карт та схем;
- адміністрування пристроїв в мережі;
- пошуку та діагностики;
- веб-сервіси для розробників додатків;
- аналітика та відслідкування в реальному часі;
- організація командної роботи над мережею;
- засоби калібрування.

Компанія Sonarax пропонує новий підхід до внутрішньої навігації на основі ультразвуку — звукових хвиль, які не чує людське вухо, проте мікрофони пристроїв все ще можуть розпізнавати такі сигнали. Наразі продукт тільки починає свій розвиток. Окрім навігації, існує багато застосувань, де ультразвукова комунікація може стати в нагоді. Публічно доступними є демонстрації взаємодії пристроїв.

Австрійська компанія indoo.rs була викуплена гігантом геосервісів Esri та стала частиною продукту ArcGis Indoors. Основною технологією є Bluetooth Low Energy. Ними розроблений повнофункціональний додаток з функціями аналітики та адаптивного калібрування карт користувачами системи SLAM Crowd Engine.

Ще один гравець ринку IndoorAtlas має на меті поєднати майже всі існуючі технології позиціонування в один продукт. Наразі підтримується геомагнітне, інерційне, GPS, Wi-Fi, маякове позиціонування. Точність за оцінками самої компанії складає 1-3 метри в режимі пішохідного переміщення («відстеження точки» — англ. blue dot). Основні функції: пошук шляху, мітки на об'єктах, розпізнавання поточного поверху та його зміни, кросплатформовість, контекстні повідомлення користувачам.

Таким чином, бачимо, що дуже поширеним є позиціонування, засноване на бездротових маяках, що випромінюють сигнал. Багато компаній розробили власний набір розробки SDK (Software Development Kit) для мобільних додатків. В таблиці 2.3 надається короткий порівняльний опис таких комерційних систем.

Таблиця 2.3. Порівняння SDK навігації

Компанія		Navigate	Estimate	Insiteo	Steerpath	Infsoft	Accuware	Atlas
Можливості	Android	+	-	+	+	+	+	+
	iOS	+	+	+	+	+	+	+
	Точність, м	1	1	2	2	1	2	1
	SLAM	+	+	-	-	-	-	-
	Wi-Fi	+	-	-	-	+	+	+
	Запис карти покриття	+	-	-	-	-	+	-

Таблиця 2.3 (продовження)

Компанія		Navigine	Estimote	Insiteo	Steerpath	Infsoft	Accuware	Atlas
Можливості	Складність налаштування	Вис.	Низ.	-	-	Сер.	Вис.	Вис.
	Робота офлайн	-	+	+	-	-	-	-
	Підтримка навігації між поверхами	+	-	-	+	+	+	+
	Режим пошуку шляху	+	-	+	-	+	-	+

Наведені вище програмні засоби потребують розгортання інфраструктури, а саме Bluetooth маяків. Рішення без інфраструктури, наприклад, NavVis потребують попереднього сканування приміщень лідаром — пристроєм, що знімає окрім звичайного зображення ще й відстань до точок (рисунок 2.7) [39].



Рисунок 2.7 — Автономний лідар NavVis M6

Популярним прикладом застосування маяків є аеропорт Гетвіка в Великій Британії (рисунок 2.8). За розробку рішення відповідала компанія Pointtr.

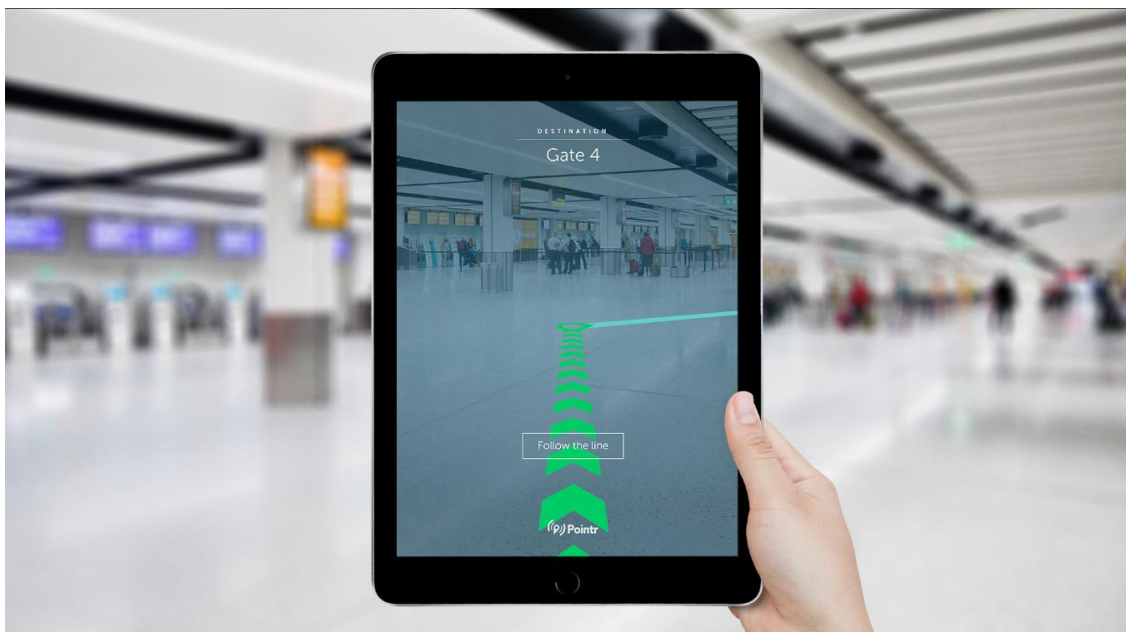


Рисунок 2.8 — Зображення процесу навігації в аеропорту Гетвік

Там розміщені 2000 маяків Bluetooth Low Energy, що живляться від батарей. Заявляється, що точність сягає 3 метрів. Розглядаючи такі фактори, як вартість маяків (10-20 доларів за одиницю), необхідність замінювати елементи живлення кожні 1-2 роки, бачимо, що використання маяків є доцільним лише в певних умовах.

2.3.2 Додатки з використанням візуального визначення позиції

Ближчими до програмного продукту, що розробляється в даній роботі, є додатки з використанням доповненої реальності.

Прикладом є демонстрація компанії Mobidev, що спеціалізується розробці прототипів. На сайті можна побачити відео демонстрації та опис двох прототипів внутрішньої навігації. Один з них заснований на фреймворку ARKit та працює лише на смартфонах компанії Apple під керуванням iOS. Інший же використовує ARCore, як і програмний продукт, розроблений в результаті даної роботи.

Зараз активно розвиваються рішення, засновані на маркерах доповненої реальності. Візуальний маркер — зображення, що розпізнається AR-системою. Вони типово використовуються як якір для розміщення контенту.

Якщо розмістити візуальний маркер в просторі на підлозі або на стіні та зберігати десь його точні координати, то з прийнятною точністю можемо прив'язати положення пристрою, який сканує маркер, до оточення.

Чим більше часу проходить від моменту синхронізації, тим більшою стає накопичена похибка вимірювання. Тому потрібно час від часу знову синхронізуватись з маркером, щоб розмір помилки не перевищував допустимі значення. З досвіду, AR-маркери мають бути на відстані близько 50 м один від одного.

2.4 Обраний підхід до підсистеми визначення положення

Отже, бачимо, що більшість технологій, на яких створюються системи навігації, потребують розгортання спеціальних пристроїв, що деяким чином взаємодіють з середовищем, в яке вони вбудовуються.

Дослідження показали, що користувачам для пішохідної навігації потрібна така точність, щоб похибка не становила більше 10% пройденого шляху [40]. Так, наприклад, якщо відстань до кімнати, в яку потрібно потрапити, складає 50 м, то похибка повинна бути не більше 5 м.

Як можна отримати таку точність без використання додаткових пристроїв, окрім смартфона? Як відповідь на це питання пропонується розширена система візуально-інерційної одометрії.

Метод SLAM забезпечує візуальну складову визначення позиції. Використання інерційних датчиків покращує точність отриманих результатів. Технологія ARCore від Google та аналоги надають розробникам саме таку систему.

На даному етапі відслідковується позиція пристрою відносно точки ініціалізації процесу локалізації. Виникає наступна проблема — потрібний метод для перетворення локальних координат пристрою в такі, щоб позиція в реальному світі відповідала позиції віртуального аналогу смартфона на сцені, що являє собою модель поверху будівлі.

Можна застосувати технології близької дії, такі як радіочастотна ідентифікація та NFC, проте в такому разі потрібні спеціальні картки або мітки, що кодують в собі інформації про точку. Так як камера активно використовується в процесі

позиціонування, має сенс використовувати оптичні методи кодування, такі як маркери.

Існує два типи маркерів — натуральні та синтетичні. Другі застосовують частіше, так як вони мають значні переваги в якості розпізнавання та інформації, що надають [41].

Найпростішим з них є статичний маркер з широкою рамкою. Це чорно-біле зображення з чіткими контурами, яке розпізнається досить легко по ключовим точкам та не потребує для цього багато обчислювальної потужності. Такі зображення розпізнаються навіть за умов поганого освітлення, цим самим мінімізують ймовірність відмови програми, що базується на них. Яскравим прикладом є Ніро з комп'ютерної бібліотеки ARToolkit.

Наступним кроком еволюції маркерів є двовимірні штрих-код шаблони. Вони кодують в собі деяку інформацію. Загальна структура залишається такою самою, якщо порівнювати два маркери між собою. Змінюється лише порядок елементів або їх колір.

Так, наприклад, SDK Vuforia включає в себе систему маркерів VuMark для створення брендovаних маркерів. Рисунок 2.9 наводить приклади таких зображень.



Рисунок 2.9 — Брендovані маркери VuMark

Як бачимо, хоч стиль кожного з них відрізняється, вони містять спільні особливості, такі як присутність зони, насиченої геометричними примітивами.

Проте з великої відстані маркери з дрібними елементами не можуть бути розпізнані. Тому часто відкидають стилізацію, залишаючи лише контрастні об'єкти.

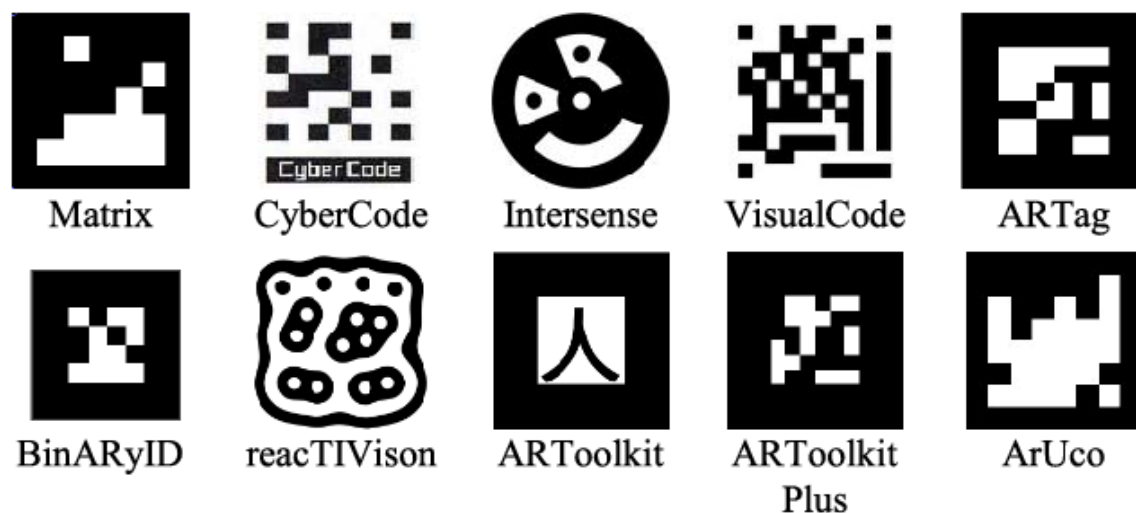


Рисунок 2.10 — Приклади простих синтетичних маркерів

Дуже важливим є таке розміщення складових частин, щоб мінімізувати помилки розпізнавання при різноманітних поворотах та згинаннях маркерів та затіненні його частин. Складені різноманітні словники положення елементарних частин на зображенні. Дослідження показують, що одним з найкращих є ArUco — розробка групи вчених іспанського університету Кордобі [42]. Алгоритм розпізнавання є досить простим та його імплементація входить до пакету OpenCV.

Алгоритм включає такі кроки [43]:

- Визначення кандидатів в маркери. Зображення аналізується на присутність квадратних фігур. Починається з адаптивного порогування, щоб виділити маркери. Відфільтровуються занадто малі, не квадратні, не закриті контури.

- Виділення бітів маркера. Спочатку відбуваються перспективні перетворення, щоб привести зображення маркера до канонічного вигляду. Методом бінарізації Оцу виділяються чорні та білі зони, що перетворюються в бітовий вигляд.

- Нарешті, біти аналізуються, щоб дізнатися чи належать вони до словника чи ні. За потреби застосовуються техніки корекції помилок розпізнавання.

У даній роботі обрано саме маркери ArUco для кодування ідентифікаторів точок прив'язки до системи координат схеми через простоту, зрілість, ефективність розпізнавання та наявність готових бібліотек розкодування.

Отже, система позиціонування для мобільного додатку внутрішньої навігації з використанням доповненої реальності, що задовольняє поставленим вимогам точності та практичності, має використовувати метод SLAM та візуально-інерційну одометрію як його частину, синтетичні маркери ArUco. Техніка позиціонування — аналіз сцени. Технологія — гібридна, поєднання навігаційного обчислення та оптичної системи.

Висновки до розділу

Таким чином, в розділі описані теоретичні основи систем внутрішньої навігації. Проведений аналіз існуючих комерційних рішень навігації в будівлях та підходів до організації таких систем.

Можна зробити висновки, що наразі зрілими компаніями в переважній більшості застосовуються методи, що вимагають розміщення інфраструктури — деяких пристроїв. Проте вартість їх впровадження та витрати на експлуатацію в поєднанні з недостатньою точністю часто переважають отримані вигоди. Перспективними є AR-методи навігації. Раніше вони були недоступними через обмеження продуктивності мобільних пристроїв.

Був запропонований підхід на базі синтезу доступних технологій доповненої реальності: реалізація методу SLAM та синтетичні маркери ArUco. Науковою новизною є удосконалення СВП через поєднання візуально-інерційної одометрії з синтетичними маркерами, що призвело до можливості розробки системи локалізації без розміщення дорогих пристроїв приймання чи передачі сигналів.

3 РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ

У розділі описуються та обґрунтовуються рішення щодо обраних засобів та технологій для розробки програмного продукту. Надається опис додатку в цілому та конкретно підсистеми позиціонування з точки зору розробника програмного забезпечення.

3.1 Засоби розробки

Під час розробки програмного продукту використовуються наступні засоби:

- Редактор Unity3D версії 2019.1.10;
- Visual Studio 2019 для редагування коду скриптів;
- ARCore 1.13.

Цільовою платформою є мобільні пристрої під керуванням операційної системи Android версії 7.0 та вище, що входять до списку підтримуваних пристроїв Google ARCore. З детальним списком можна ознайомитись на сайті документації ARCore.

Технологія ARCore працює також на смартфонах компанії Apple, проте такий випадок не був протестований.

3.1.1 Редактор Unity

Unity — багатоплатформовий засіб для розробки 2D та 3D програм [44]. Сам засіб запускається на ОС Microsoft Windows та Apple MacOS, хоча дозволяє розробляти застосунки, що працюють на різноманітних платформах, від класичних персональних комп'ютерів та ігрових консолей Sony PlayStation, Microsoft Xbox до мобільних пристроїв та навіть веб-додатків. Інструмент Unity серед конкурентів вирізняється присутністю середовища розробки візуального типу, кросплатформової компіляції і модульної системи компонентів. До недоліків відносять появу труднощів при підключенні зовнішніх бібліотек через застосування середовища виконання Mono.

Редактор Unity включає інтерфейс з підтримкою принципу «перетягни та відпусти», який нескладно налаштовується. Він є модульним, включає в себе різні

вікна. Як скриптові мови програмування використовуються C# та модифікована версія JavaScript. Ще нещодавно також підтримувався Boo (діалект Python), але його прибрали в 5-й версії через низьку популярність.

Проект Unity складається з незалежних сцен — окремих контейнерів, що містять свої налаштування об'єктів, їх структуру. Одночасно завантажена може бути лише одна сцена, можна їх змінювати. Такі файли можуть містити в собі ігрові об'єкти (моделі) та об'єкти функціоналу, що не мають моделі та ніяк не відображаються. Об'єкти функціоналу потрібні для забезпечення логіки додатку, налаштування, передачі даних та інших скриптових потреб. Всі об'єкти, в свою чергу містять прив'язані компоненти. Також у об'єктів є неунікальна назва (дозволяється, щоб два чи більше об'єктів мали однакову назву), може бути шар, на якому він відмальовується і мітка (тег). Компонент під назвою Transform є обов'язковим для всіх елементів сцени — він зберігає в собі координати позиції, Ейлерові кути повороту і коефіцієнти масштабу об'єкта. У об'єктів з видимими елементами також присутній компонент відрисовування сітки за замовчуванням, що робить його модель видимою у світовому просторі.

До об'єктів можна застосовувати колізії (в Unity колайдери — collider), яких існує декілька типів.

Безкоштовна версія Unity має деякі обмеження, але в її межах є можливість поширювати гри за умови, що щорічний дохід з гри не перевищує 100 000 доларів.

Список пакетів, що застосовані в проекті Unity:

- Android logcat — для перегляду логів з пристрою Android, потрібний для відлагодження, деякі проблеми проявляються лише безпосередньо на пристроях;
- JSON.Net для Unity — портована версія бібліотеки Newtonsoft.Json з модифікаціями для Unity для серіалізації та десеріалізації файлів формату JSON;
- XR Legacy Input Helpers — забезпечує інтерфейс для взаємодії Unity з AR пристроями, необхідний для роботи ARCore.

3.1.2 Платформа ARCore

ARCore — платформа Google для створення додатків розширеної реальності. Використовуючи різні API, ARCore дозволяє вашому телефону розпізнати своє

оточення, зрозуміти світ та взаємодіяти з ним. Деякі API доступні для Android та iOS, щоб зробити можливим спільний AR досвід [45].

ARCore використовує три ключові можливості для інтеграції віртуального контенту з реальним світом, що спостерігається через камеру вашого телефону:

- відстеження руху дозволяє телефону зрозуміти та відстежувати його положення відносно світу.

- розуміння оточення дозволяє телефону визначати розміри та розташування всіх типів поверхонь: горизонтальних, вертикальних і розміщених під кутом, таких як земля, журнальний столик або стіни.

- оцінка світла дозволяє телефону оцінити поточні умови освітлення навколишнього середовища.

По суті, ARCore займається двома задачами: відстежує положення мобільного пристрою під час його руху та будує власне розуміння реального світу. Технологія відстеження руху ARCore використовує камеру телефону для виявлення точок інтересу, що називаються ознаками та відстежує, як ці точки рухаються з часом. За допомогою поєднання руху цих точок та показань з інерційних датчиків телефону ARCore визначає як положення, так і орієнтацію телефону під час руху в просторі.

Розуміння ARCore реального світу дозволяє розміщувати предмети, примітки чи іншу інформацію таким чином, що вони непомітно інтегруються з реальним світом. Ви можете розмістити кошеня, що дрімає, на кутку свого журнального столика або анотувати картину біографічною інформацією про художника. Відстеження руху означає, що ви можете пересуватися та переглядати ці предмети з будь-якого кута, і навіть якщо ви повернетесь та вийдете з кімнати, коли повернетесь, кошеня чи анотація будуть саме там, де ви їх залишили.

Існує два типи додатків ARCore: AR Required та AR Optional. Обов'язковий режим AR означає, що додаток не можна використовувати жодним чином на телефонах без підтримки можливостей доповненої реальності. Опціональний режим дозволяє запускати програми на тих пристроях, де не підтримується ДР, тобто вони повинні мати корисний функціонал не пов'язаний з ARCore.

Класифікація впливає на поширення додатку через магазин Google Play Store. Додатки з обов'язковим AR неможливо встановити на пристрої, які офіційно не підтримують ARCore. В цьому разі система автоматично підтримує необхідну

залежність ARCore під назвою «Google Play Services for AR». Якщо її немає, то Play Store без втручання користувача встановить її або ж у випадку застарілої версії оновить сервіси.

Мобільні застосунки з необов'язковим режимом ДР повинні робити декілька перевірок під час роботи: що пристрій підтримується, що встановлена остання версія «Google Play Services for AR». Тільки якщо обидві умови виконуються, то може застосовуватись функціональність доповненої реальності. В іншому випадку рекомендується вивести повідомлення користувачеві та запропонувати оновити чи встановити залежність.

Розроблюваний додаток не має жодного сенсу без використання ARCore, тому він працює в режимі AR Required. В Unity режим налаштовується в меню Edit > Project Settings > ARCore > ARCore Required у вигляді прапорця.

Отже, описані інструментальні засоби розробки, що використані при створенні та відлагодженні програмного продукту, бібліотеки та їх функції в розробленій системі.

3.2 Структура додатку

Ієрархічну структуру розробленої системи описує рисунок 3.1.

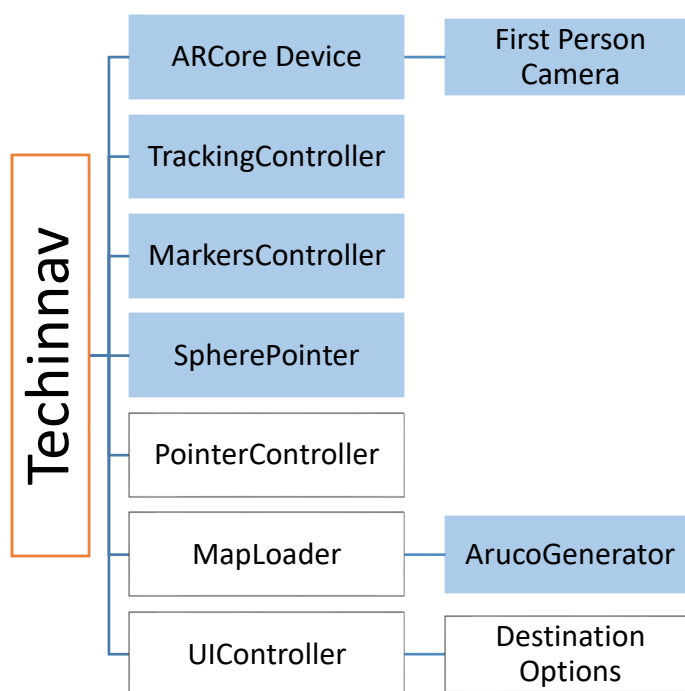


Рисунок 3.1 — Загальна структура розробленої системи

На ньому зображені об'єкти сцени Unity. Підсистема позиціонування (виділена на рисунку 3.1 синім) складається з наступних частин:

- ARCore Device — об'єкт, який зберігає стан сесії ARCore. Також він містить її початкову конфігурацію.
- First Person Camera — головна камера сцени. Через неї проектується зображення, що бачить користувач на екрані.
- TrackingController — контролер, який керує відстеженням позиції.
- SpherePointer — об'єкт, що символізує користувача в просторі.
- MarkersController — контролер, що відповідає за розпізнавання маркерів ARUco.

Розглянемо детальніше кожний з них.

3.3 Об'єкт сесії ARCore

Для функціонування програми потрібна наявність об'єкту з компонентом ARCore Session, який зазвичай називається ARCore Device. Скрипт сесії служить точкою входу та ініціалізації ДР. Також він містить у собі всі налаштування ARCore.

Перш за все, визначається з якої камери знімається зображення: з задньої чи фронтальної. По замовчуванню обрана задня камера, так як вона має кращу якість зйомки та знімає те ж, що бачить користувач. Проте в деяких випадках доцільно вибрати передню камеру, для тих додатків, що оперують знімками користувача (наприклад, Snapchat).

Наступною властивістю налаштування AR Core Session є об'єкт SessionConfig. В ньому можна включити чи виключити деякі додаткові опціональні можливості:

- розпізнавання площин;
- оцінка освітленості сцени;
- хмарні якорі;
- доповнені зображення;
- режим фокусу камери (фіксований чи автоматичний фокус);
- доповнені обличчя.

Увімкнене розпізнавання площин намагається виділити зі сцени плоскі геометричні об'єкти. Підтримуються режими виключно горизонтальних площин, тільки вертикальних та всіх. Площини можна використовувати для прив'язки до них віртуальних предметів. Одночасно розпізнаними можуть бути багато площин, розробник додатку сам обирає потрібну йому на основі деяких критеріїв, наприклад, висота, близькість, площа. Рисунок 3.2 демонструє розпізнання площини підлоги.



Рисунок 3.2 — Візуалізація розпізнаної горизонтальної площини

Також ARCore підтримує оцінку освітленості сцени. Це дозволяє відмальовувати віртуальні об'єкти, які підлаштовуються по освітленню та тіням до реальних, створюючи більший ефект нерозривності реального та віртуального. Ще одним застосуванням є додавання логіки на основі інтенсивності освітлення. Так, ARCore включає демонстрацію, в якій відносно віртуальної моделі лева починається анімація страху, коли різко змінюється інтенсивність світла, наприклад, при вимкненні кімнатних джерел світла.

Люди підсвідомо аналізують зображення на предмет неспівпадіння і якщо на ньому присутній об'єкт, який не створює тіні або вирізняється яскравістю, вони відчують, що предмет не належить сцені, хоча часто не можуть пояснити чому саме. Рисунок 3.3 показує принципи оцінки освітленості.

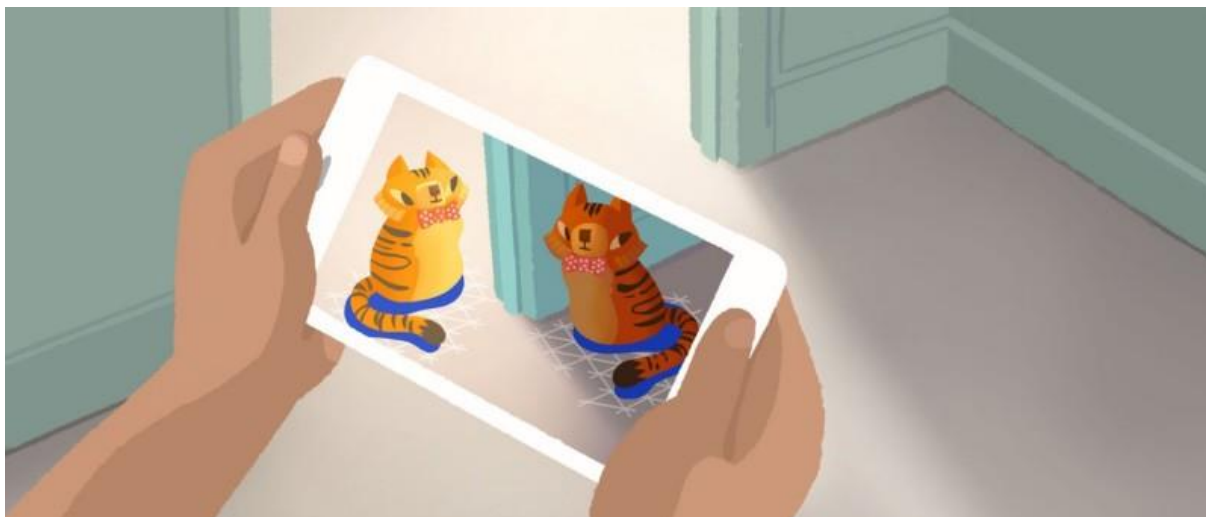


Рисунок 3.3 — Оцінка освітленості віртуальних об'єктів

Існує три режими налаштувань цього параметра:

- виключений;
- на основі інтенсивності зовнішнього освітлення (`AmbientIntensity`);
- `EnvironmentalHDR` — враховуються головне джерело направленої світла, інші джерела зовнішнього освітлення, відблиски.

Хмарні якорі використовуються для кооперативної взаємодії користувачів через мережу Інтернет. Вони дозволяють одночасно переглядати одні й ті ж віртуальні предмети з різних точок зору та різними користувачами, що знаходяться в одному фізичному просторі. Така функція дозволяє досить легко створювати додатки з елементами багатокористувацьких ігор.

Якорі зберігаються протягом 24 годин з моменту завантаження на сервери та обробки. Наразі розробляються постійні якорі, час життя яких буде більшим, але на момент написання роботи вони ще не були випущені.

Для створення хмарного якоря на сервери Google завантажується зображення з камери за останні 30 секунд. Тому гостро стоїть питання приватності, користувачі мають погодитись з такими умовами.

Функція доповнених зображень дозволяє розробляти додатки, які реагують на завчасно визначені двовимірні зображення в користувацькому оточенні, наприклад постери, газети, книжки, пакування продуктів тощо.

Починаючи з версії 1.9 ARCore підтримує розпізнавання зображень, що рухаються, наприклад, реклама на автобусі в русі або зображення, яке тримає людина в руках.

Розробник додає набір зразкових зображень. ARCore шукає їх реальні копії в кожному кадрі. Встановлений ліміт в 20 одночасно відстежуваних зображень. Всього ж бази зразків може бути до 1000 елементів. Варто пам'ятати, що чим більше зразків, тим більше часу витрачає система на обробку кожного кадру. Тому варто тримати кількість зразкових зображень мінімально можливою.

На початку відстеження зображення в додаток надається оцінка розміру реального зображення, його позиція та орієнтація в просторі. Оцінки можуть уточнюватись в процесі отримання системою нової інформації.

Якщо на момент компіляції відомі зразкові зображення, то необхідно додати їх в базу даних, використовуючи утиліту `arcoring`. Попередньо вона перевіряє придатність зображення, оцінюючи її від 0 до 100. Критеріями оцінки є наявність ознак, що повторюються, контрастність, роздільна здатність. Рекомендується оцінка як мінімум в 75 балів.

Функціонал доповнених обличь ARCore дозволяє додатку автоматично ідентифікувати різні зони розпізнаного обличчя та застосовувати їх для накладання текстур та моделей так, щоб вони співпадали з контурами конкретного обличчя.

В ієрархію об'єкту `ARCoreDevice` входить камера `First Person Camera`. Все, що вона знімає дублюється на екран мобільного пристрою. Дочірніми об'єктами камери є елементи користувацького інтерфейсу, але вони в даній роботі розглядатися не будуть, бо є частиною іншої підсистеми програми.

Дуже важливо під час запуску мобільного застосунку перевіряти необхідні умови для ініціалізації сесії. В додатку за це відповідає `TrackingController`. Властивість `Status` статичного класу `Session` є основним індикатором стану ARCore. Вирізняють наступні можливі стани:

- `None = 0` — сесія ще не була ініціалізована, потрібно зачекати;
- `Initializing = 1` — сесія ініціалізується на даний момент;
- `Tracking = 100` — ARCore працює та відслідковує пересування смартфона;

- `LostTracking` = 101 — сесія втратила відстеження, намагається відновитися;
- `NotTracking` = 102 — відстеження призупинене;
- `FatalError` = 200 — сталась внутрішня непоправна помилка в `ARCore`;
- `ErrorApkNotAvailable` = 201 — сервіси Google Play Services for AR відсутні на мобільному пристрої або версія застаріла;
- `ErrorPermissionNotGranted` = 202 — не надані потрібні дозволи до ресурсів пристрою, наприклад, до камери та датчиків;
- `ErrorSessionConfigurationNotSupported` = 203 — не надані налаштування `ARCore` або обране таке їх поєднання, що не підтримується.

Як бачимо, сесія обов'язково знаходиться в одному з наведених статусів і переходить в одного стану в інший. Статуси поділені на декілька груп: початкові (код менше 100), нормальної роботи (від 100 до 200) та помилкові (від 200 і більше), при яких подальша робота неможлива.

Зазвичай, сесія проходить через статуси 0 та 1 до статусу 100. В разі якщо телефон занадто швидко рухається або в кадр не потрапляють виразні точки-орієнтири, статус тимчасово задається як 102 — втрата відслідковування.

Рекомендується стимулювати користувачів не робити різких рухів та не наводити камеру телефону на одноманітні поверхні без виразних точок-орієнтирів.

3.4 Відстеження позиції

Відразу після відкриття програми та після закінчення початкового налаштування `ARCore` запускається процес безперервної локалізації пристрою в навколишньому середовищі. `ARCore` ініціалізується з початковою позицією (0, 0, 0) та в напрямку (0, 0, 0).

З кожним кадром `ARCore` будує хмару точок, за допомогою яких прив'язується до середовища. Також з частотою 1 кГц відслідковуються покази сенсорів телефону, перетворюються і інтегруються. Так, з показів акселерометра шляхом інтегрування дізнаємось відстань, на яку перемістився пристрій. `ARCore` записує локальну позицію

пристрою у властивість Pose статичного класу Frame. Клас Pose складається з тривимірного вектора позиції Position та кватерніона Rotation.

Для кінцевого користувача переміщення відображається як відповідне зміщення об'єкта SpherePointer, що показаний в центрі на рисунку 3.4. В інтерфейсі приміщення показується як тривимірне, проте ми оперуємо при синхронізації та інших перетвореннях двовимірними точками, що є проекціями на площину xOz.

Висота вказівника над рівнем підлоги є фіксованою та складає 0.5 одиниць.

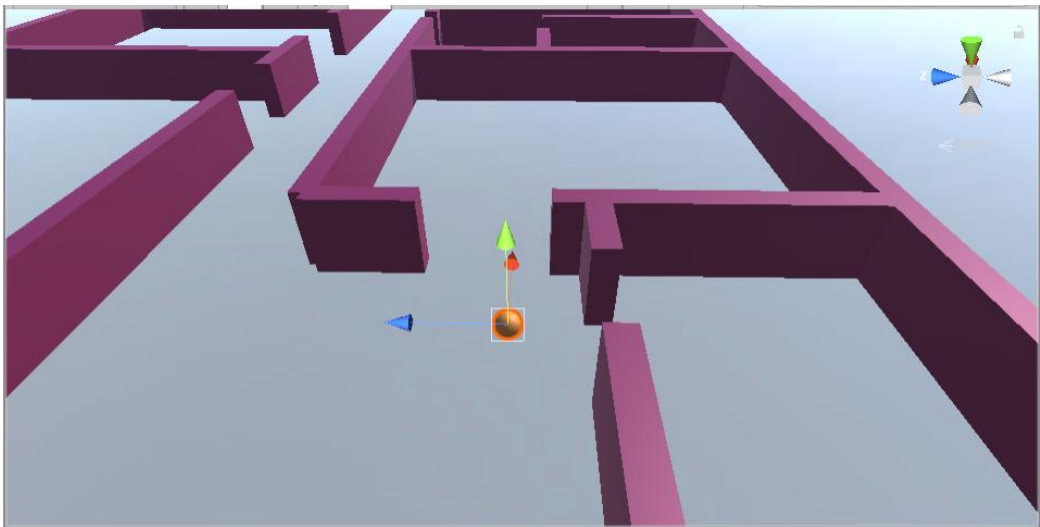


Рисунок 3.4 — Вигляд сфери-вказівника

Якби користувач завжди запускав додаток в точці, що є початком системи координат віртуальної сцени та в напрямку на північ, то тільки можливостей ARCore було б достатньо. Але це є неприйнятним з точки зору зручності, тому потрібно якимось чином задати відповідність позиції ARCore та на схемі. Виявляється, що не можна змінити початкову позицію пристрою в ARCore, вона завжди при старті становить (0, 0, 0), так само і напрям. Для приведення цієї позиції до шуканої використовуються маркери ArUco. Знаючи точку розташування віртуального маркеру, його орієнтацію та відносну позицію реального, ми можемо знайти деякі параметри, що перетворюють систему координат ARCore в подібну до Unity. Тоді, нова позиція сфери вираховується за формулою (3.1):

$$\vec{x} = q\vec{x}_0q^{-1} + \vec{\Delta x}, \quad (3.1)$$

де q — кватерніон, що повертає систему координат ARCore на α градусів навколо осі OY;

Δx — постійний відступ від початку координат;

\vec{x}_0 — значення Frame.Pose.Position.

Рисунок 3.5 схематично зображує ці параметри в процесі визначення глобальної позиції.

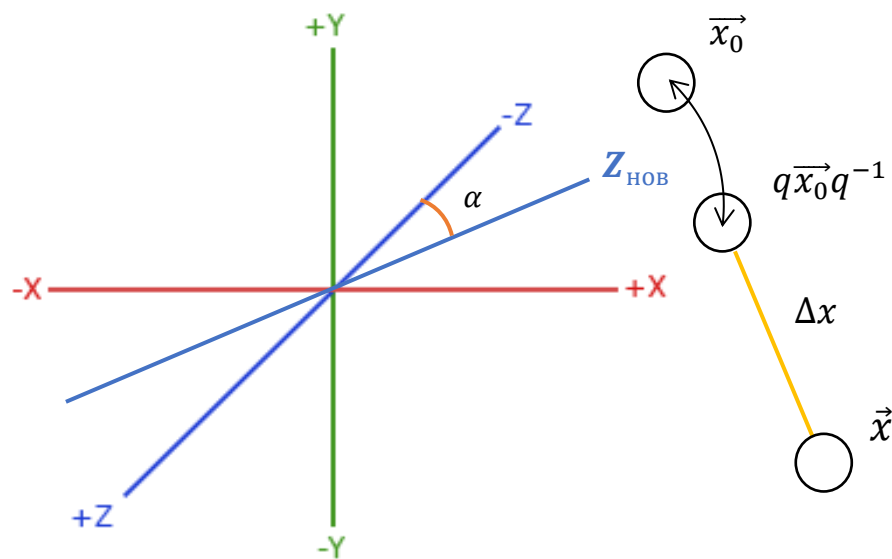


Рисунок 3.5 — Визначення позиції сфери-показника

Указані параметри знаходяться на початку роботи програми та під час кожної синхронізації з маркером. При запуску додатку Δx встановлюється в значення, що дорівнює позиції віртуального показника позиції. Тобто точці початку координат ARCore відповідає задана в редакторі позиція сфери-вказівника.

Розглянемо алгоритм знаходження параметрів. Припустимо, що маркер розпізнався та ми знаємо його координати в обох системах координат: в локальній ARCore та світовій Unity. Довжина базисних векторів співпадає в обох системах. Нехай \vec{y}_{cx} — позиція маркера на схемі, \vec{x}_{AR} — позиція смартфона в системі координат ARCore, \vec{x}_y — оцінка вектора напрямку від телефону до маркера, Z_{AR} — напрям осі Z ARCore (рисунок 3.6).

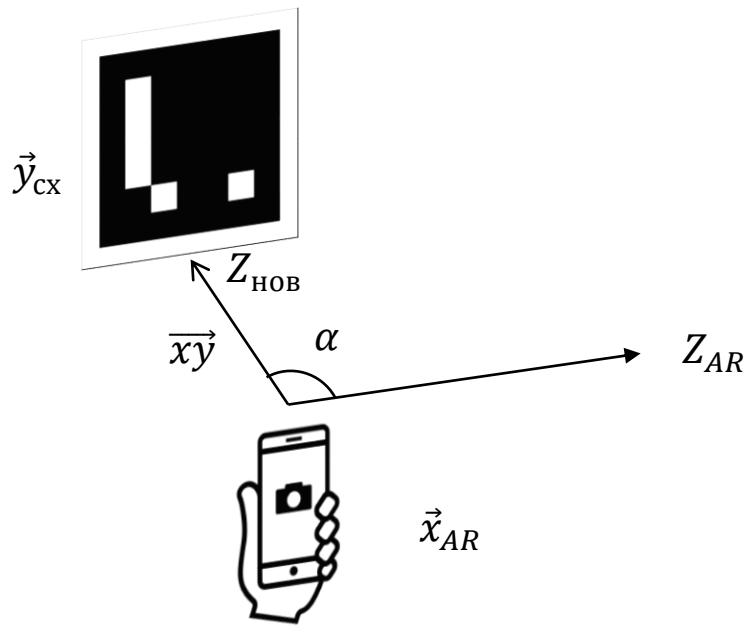


Рисунок 3.6 — Складові визначення параметрів синхронізації

Тоді, поворот системи координат визначається як кут між векторами $\vec{x}\vec{y}$ та одиничним базисним вектором $\vec{z} = \overrightarrow{(0, 0, 1)}$. Далі визначимо позицію смартфона у світовій системі координат Unity:

$$\overrightarrow{x_{cx}} = \overrightarrow{y_{cx}} - \vec{x}\vec{y} \quad (3.2)$$

Та зсув відносно початку координат ARCore:

$$\overrightarrow{\Delta x} = \overrightarrow{x_{cx}} - q\overrightarrow{x_{AR}}q^{-1} \quad (3.3)$$

Отримані значення записуються у властивості `CameraState.RotationDrift` та `PointerState.PositionDrift`. Надалі, позиція вказівника у віртуальному просторі має співпадати з реальним станом речей.

3.5 Генерація та розпізнавання маркерів

Маркери `ArUco` становлять собою сіткоподібні чорно-білі контрастні зображення, що кодують в собі число-ідентифікатор. Вони складаються з таблиці

$N \times N$ квадратів, оточених чорною внутрішньою та білою зовнішньою рамками. Комбінація внутрішніх квадратів власне і зберігає в собі дані.

Варто відзначити, що маркер може бути повернутий в просторі, тому процес розпізнавання повинен враховувати даний факт. Словник маркерів — набір маркерів, що застосовується в додатку. При цьому ідентифікатор маркера — порядок в словнику, а не десяткове представлення внутрішньої матриці, як можна було б подумати.

На рисунку 3.7 зображений приклад маркера розмірності 7 (всередині 7 рядків та 7 стовпчиків). В конкретному словнику він має ідентифікатор 1.

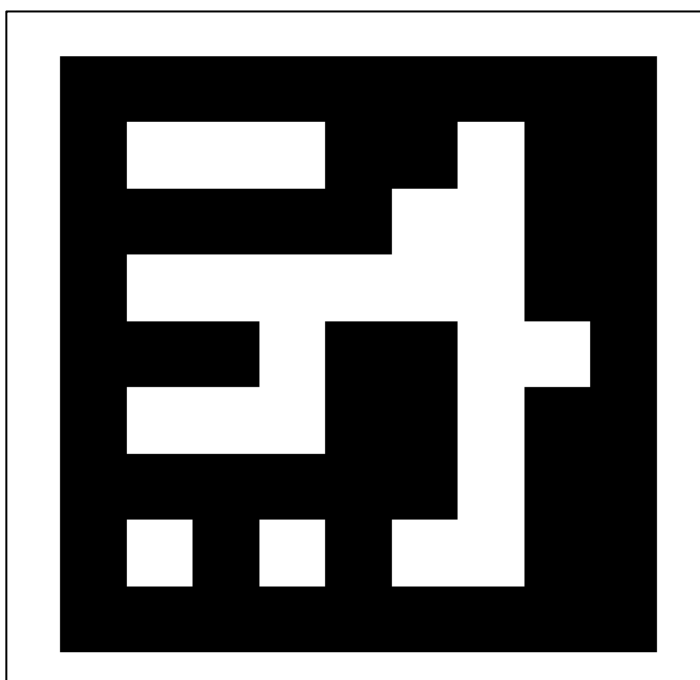


Рисунок 3.7 — Приклад маркера синхронізації позиції

Білі квадрати означають 1, а чорні — 0. Враховуючи це, вище наведений маркер має наступне бінарне представлення: 1110010 0000110 1111110 0010011 1110010 0000010 1010110. В словниках, що застосовуються в програмі, цей маркер зберігається як [228, 27, 241, 62, 64, 171, 0] — масив 8-бітних чисел, що доповнюються нулями справа.

При розробці схеми поверху маркери відмічаються дугами, що анотовані рядком шаблону MARKER:N, де N — ідентифікатор маркера. Зазвичай маркери розміщують на стінах біля входів на поверх. Під час імпорту схеми формату DXF в

Unity генеруються відповідні зображення та додаються в базу даних ARCore Augmented Images в фоновому режимі. За цей процес відповідає компонент ArucoGenerator. Він має два публічних методи: AddToImageDatabase та GenerateMarker для додавання до бази зразків ARCore та генерації зображення відповідно.

Для генерації маркерів використовується компонент Unity Texture2D, що надає можливість оперувати двовимірними текстурами в процесі виконання. Алгоритм генерації є наступним:

- завантажити в пам'ять словник маркерів;
- знайти послідовність байтів, що відповідає ідентифікатору;
- операторами побітового зсуву виділити матрицю пікселів;
- створити Texture2D розміром $(N + 4) \times (N + 4)$;
- заповнити рамки: зовнішню білим кольором, внутрішню — чорним;
- заповнити внутрішні пікселі значеннями матриці з кроку 3;
- збільшити текстуру до розміру 1024×1024 методом найближчого сусіда;
- перетворити текстуру в PNG зображення.

Отримане зображення додається в базу даних ARCore. Варто відзначити, що процес додавання займає певний час (близько 20 мс, що досить багато для одного кадру). Якщо на схемі розміщено багато точок-маркерів, то може з'явитись затримка під час її завантаження. Тому рекомендується робити додавання зразків асинхронним, таким, що не блокує основний потік графічного інтерфейсу, щоб користувач не мав відчуття втрати реакції на його дії.

Відразу після закінчення ініціалізації ARCore починає шукати в кожному кадрі ознаки маркерів. За це відповідає MarkersController. Це відбувається викликом метода Session.GetTrackables, що записує в попередньо виділений список перелік всіх об'єктів певного типу, які відстежуються ARCore. В даному випадку метод викликається з параметром-типом AugmentedImage, тобто отримуємо всі розпізнані доповнені зображення в кадрі. Навіть якщо зображення тимчасово зникло з поля зору камери смартфона, його остання позиція все ще відстежується. Під кожне зображення відразу створюється якір. Якорі гарантують, що об'єкти, прив'язані до них будуть

знаходиться в одному і тому ж місці, цим самим допомагають підтримувати ілюзію злиття віртуальних та реальних об'єктів на одній сцені.

В класі `AugmentedImage` є дві корисні властивості, які називаються `TrackingState` та `TrackingMethod`. Їх значення показують статус відстеження конкретного розпізнаного зображення.

Стан відстеження `TrackingState` може набувати наступних значень:

- `Tracking` — сутність активно відстежується;
- `Paused` — відстеження призупинено, але може бути відновленим;
- `Stopped` — `ARCore` повністю зупинив відстеження сутності і більше

ніколи не продовжить процес.

Властивість `TrackingMethod` має одне з таких значень:

- `FullTracking` — зображення відстежується в даний момент часу;
- `LastKnownPose` — наразі зображення відсутнє в кадрі, проте продовжується відслідковуватись, базуючись на останньому значенні позиції;
- `NotTracking` — не відстежується взагалі.

Ми можемо бути впевненими, що маркер відстежується тоді і тільки тоді, коли стан відстеження дорівнює `Tracking`, а метод — `FullTracking`. Як тільки реальна копія маркера потрапить в об'єктив та властивості `AugmentedImage` набудуть зазначених значень, згенерується подія, що знайдений об'єкт з ідентифікатором `N`. Опрацьовуючи таку подію, визначаються параметри q та $\overrightarrow{\Delta x}$. Також в цей момент потрібно скинути сесію `ARCore` до початкового стану, щоб видалити накопичену хмару точок, так як релокалізація після повороту може бути виконана неправильно. Для цього компонент `ARCoreSession` видаляється і через 1 секунду створюється ще раз з тими ж налаштуваннями.

Враховуючи все вищесказане, можна навести діаграму залежності типів в підсистемі (рисунок 3.8). Діаграма згенерована засобами розширення `JetBrains Resharper` для `Microsoft Visual Studio` та показує взаємозалежність типів в проекті. Бачимо, що центральною структурою даних є клас `DynamicMapOptions`, в який збирається інформація про об'єкти на схемі, зокрема про маркери. Точками входу є контролери `TrackingController` та `MarkersController`. Вони представлені як компоненти, прив'язані до відповідних об'єктів на сцені.

Синіми лініями вказані загальні залежності, зеленими штрихованими — посилання через поля, фіолетовими — застосування типу як такого, що повертається.

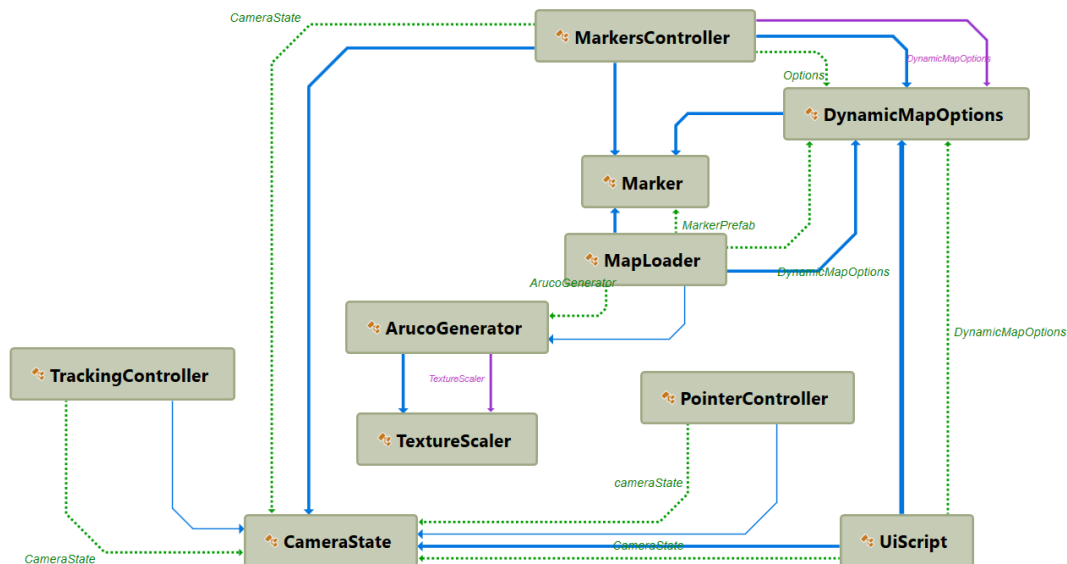


Рисунок 3.8 — Діаграма залежності типів

Більш традиційна діаграма класів виглядає наступним чином (рисунок 3.9):

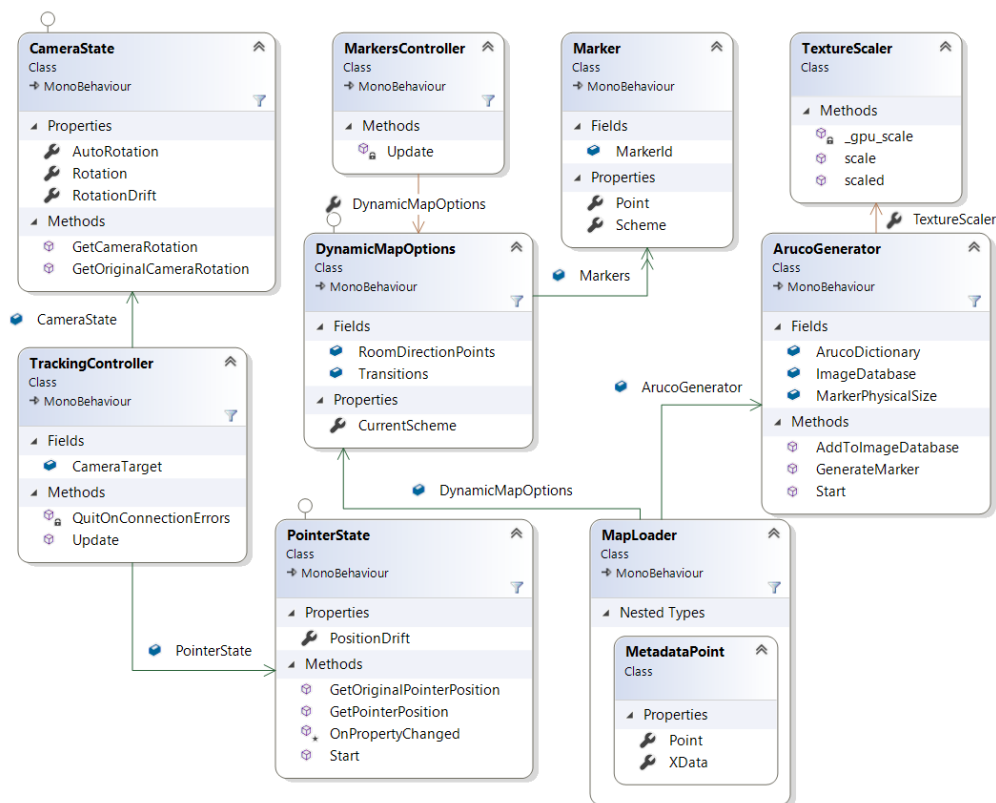


Рисунок 3.9 — UML-діаграма класів

Вона є більш детальною та зображує також деякі з наявних методів (відфільтровані вручну, вибрані тільки основні методи), а також поля та властивості. Її було створено за допомогою вбудованих засобів Visual Studio 2019.

Бачимо, що майже кожний клас успадковується від базового класу Unity для всіх компонентів — `MonoBehaviour`. Він є відправною точкою скриптів та надає методи подій життєвого циклу сцени, що називаються повідомленнями. Ось основні з них:

- `Awake` — викликається після завантаження скрипту;
- `Start` — перший кадр, коли компонент увімкнувся;
- `Update` — кожний кадр, найчастіше використовується;
- `FixedUpdate` — кожний крок фізичного двигуна, не залежить від частоти відрисуння, кожні 20 мс по замовчуванню;
- `LateUpdate` — після `Update` всіх об'єктів;
- `OnDisable` — компонент виключений;
- `OnEnable` — компонент включився.

Контролери обов'язково мають метод `Update`, який реагує на дії користувача. Класи `PointerState` та `CameraState` зберігають стан камери та сфери-вказівника відповідно та обидва реалізують інтерфейс `INotifyPropertyChanged`, що дозволяє підписуватись на зміну властивостей станів. Цим активно користується контролер користувацького інтерфейсу. В тілі кожної властивості, що підтримує реактивність, необхідно перевіряти чи змінилось значення порівняно з попереднім. Якщо змінилось, що викликається метод `OnPropertyChanged`, що запускає подію `PropertyChanged`. Всі підписники отримують лише ім'я зміненої властивості чи поля, отримати нове значення вони повинні самі.

3.6 Навігація між поверхами

Сучасні будівлі, в основному, є багатоповерховими. Виникає необхідність проводити користувачів з одного поверху на інший. В розробленому додатку використано напіваавтоматичний підхід зміни поверхів.

Кожна схема представляє собою цілий поверх або його частину. Так, 3 і 4 поверхи корпусу №5 задані однією схемою, а 5 поверх — 2 схемами, бо його частини ізольовані між собою.

В схемі приміщення задаються точки переходу між поверхами, наприклад, сходи. Кожна точка анотується назвами схем, на які вона веде. Тобто, оперуючи лише однією схемою ми можемо направити користувача до будь-якої точки інтересу.

При завантаженні програми в пам'ять додаються всі точки інтересу та назва схеми, яка їм відповідає. Маркер задає поточну схему, тобто змінюється віртуальне оточення, якщо користувач перейшов на інший поверх та синхронізувався з маркером.

У разі, якщо користувач обирає зі списку точок інтересу ту, яка розміщена на іншій схемі, додаток шукає переходи з поточної схеми до обраної. Для спрощення, на даний момент не підтримуються транзитивні переходи, тобто потрібно анотувати схему $S1$, що вона веде на $S3$, якщо існує набір відношень $S1 \rightarrow S2, S2 \rightarrow S3$.

Зі знайдених точок переходу обирається та, що знаходиться найближче. Користувачеві виводяться інструкції як пройти до неї, на скільки поверхів піднятися чи спуститися і що після цього потрібно знайти новий маркер синхронізації.

З точки зору системи позиціонування, для покращення такого випадку використання мобільного додатку можна застосовувати вбудований датчик, що вимірює атмосферний тиск — барометр. Як ми знаємо, тиск змінюється з висотою за певним законом. Тому, за допомогою показів барометра можна автоматизувати контроль висотного положення мобільного пристрою. В такому разі схеми можуть змінюватись автоматично, а синхронізація стала б необов'язковою.

Проте, на даний момент такий функціонал не підтримується. Дану ідею можна реалізувати окремо в якості покращення розробленого програмного продукту. Також гарною ідеєю, яка ще не була реалізована, є візуалізація елементів конструкції сходів, ескалаторів, ліфтів тощо. Таке покращення спростить розуміння схеми. І, наостанок, можливість вручну переключати схеми надасть можливість обирати користувацьку точку призначення при навігації між поверхами. Наразі, така функція доступна тільки на поточному поверсі.

Висновки до розділу

Таким чином, в розділі детальніше розглянута реалізація системи, її компоненти. Були наведені діаграми залежності основних модулів, обґрунтований вибір фреймворків та проектування програмних рішень.

Технологічною основою розробки підсистеми позиціонування та мобільного додатку в цілому стала платформа доповненої реальності ARCore від компанії Google. Вона надає інформацію про відносне переміщення смартфона, вміє розпізнавати зразкові зображення, оцінювати освітлення та має інші корисні можливості для застосунків з використанням AR.

Були розроблені засоби генерації маркерів ArUco, тобто перетворення масиву чисел в зображення в спеціальному форматі.

Наведені формули та принцип знаходження коефіцієнтів приведення локальної системи координат ARCore до світової Unity за допомогою маркерів.

Розроблений додаток вирізняється з-поміж конкурентів простотою налаштування та розгортання. Для його роботи потрібний тільки смартфон та роздруковані зображення маркерів синхронізації.

4 РОБОТА КОРИСТУВАЧА З ПРОГРАМНОЮ СИСТЕМОЮ

У розділі розглядається особливості впровадження та інсталяції розробленого доадтку, системні вимоги до пристроїв. Описується типовий сценарій користування мобільним додатком та основні його функції.

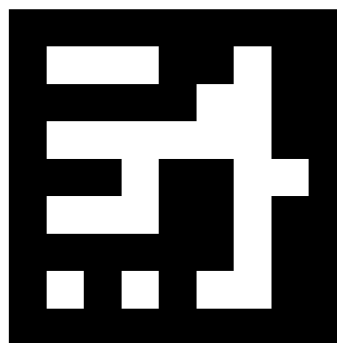
4.1 Системні вимоги та інсталяція

Наразі підтримується лише встановлення додатку на мобільний пристрій з включеним режимом розробника. Планується розміщення застосунку в магазині Google Play, що дасть змогу користувачам використовувати програмний продукт для цілей навігації в навчальному корпусі №5 Національного технічного університету України «КПІ ім. Ігоря Сікорського». Для роботи з іншими приміщеннями необхідна заміна схем поверхів та повторна компіляція.

Пропонується розміщати посилання на завантаження додатку поруч з маркерами синхронізації. Тоді користувачі, які ще не разу не користувались програмою зможуть відразу розпочати процес навігації. Рисунок 4.1 зображує те, як може виглядати відповідний постер.



Додаток
навігації



Маркер
синхронізації

Рисунок 4.1 — Приклад постера поширення додатку

У QR-кодi зашифроване посилання на завантаження пакету з програмою з сервісу Mediafire. Наразі користувач має дозволити встановлення додатків з неперевіраних джерел, але така необхідність відпаде як тільки програма буде поширюватись через авторизовані офіційні магазини.

4.2 Сценарій роботи користувача з системою

Після запуску програми користувачеві необхідно знайти в приміщенні маркер синхронізації позиції ArUco. Він виглядає таким чином, як показано на рисунку 4.1 справа (закодований ідентифікатор — 1).

Наступним кроком є сканування друкованого зображення камерою мобільного пристрою. В кожному з них закодований ідентифікатор — додатне число, що унікально ідентифікує маркер. Після цього, автоматично встановиться позиція покажчика та поворот камери, який має співпадати з реальним положенням користувача. Такий процес керується попередньо встановленою точкою на схемі, що символізує маркер. На рисунку 4.2 бачимо, що покажчик (жовта сфера) знаходить поруч з віртуальним аналогом маркера, камера повернута перпендикулярно до нього, на фоні зображення реальної копії.

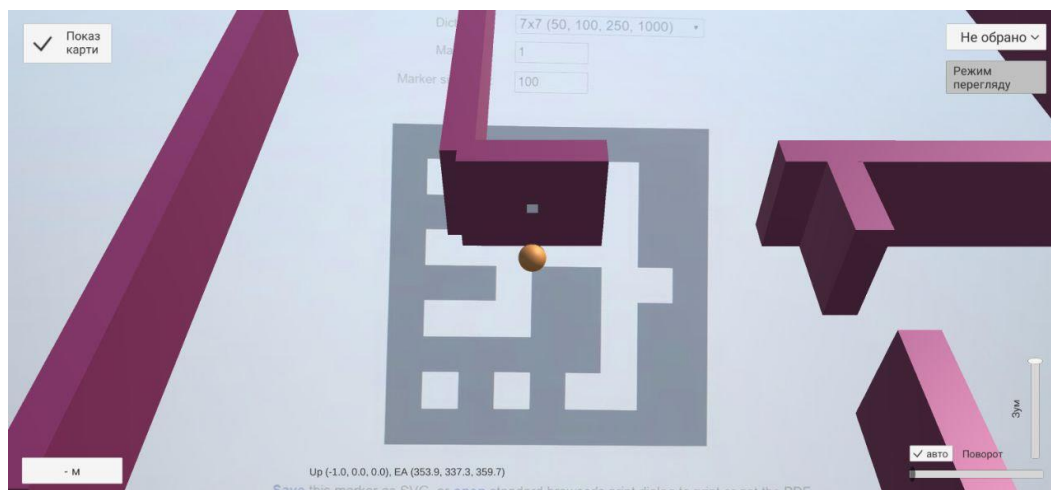


Рисунок 4.2 — Стан мобільного додатку після синхронізації

Тепер переміщення мобільного пристрою відстежується та відображається як зміщення жовтого покажчика. На даному етапі користувач задає бажану кінцеву точку навігації.

Програма підтримує декілька способів обрання точки призначення. Перший з них є більш очевидним та полягає у виборі доступних об'єктів зі списку в правому верхньому куті інтерфейсу (рисунок 4.3).

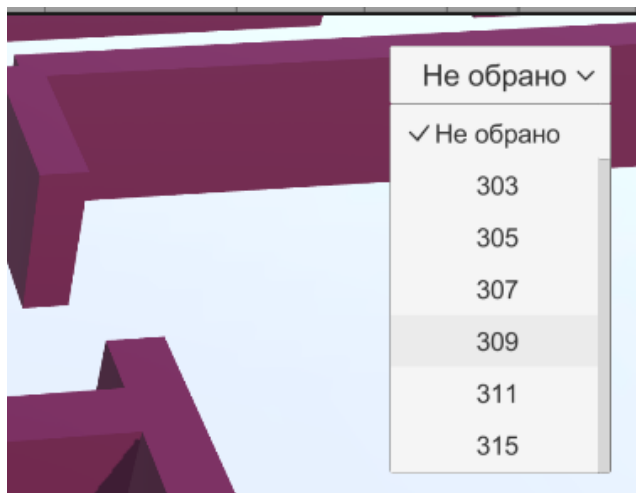


Рисунок 4.3 — Список доступних аудиторій для навігації

Альтернативою вибору однієї з помічених точок є створення власної точки призначення. Програма оброблює жест «довге натиснення» для виконання цієї дії. Необхідний час утримання пальця на одному й тому ж місці складає 1 секунду. Після цього будується траєкторія навігації саме в це місце (рисунок 4.4).

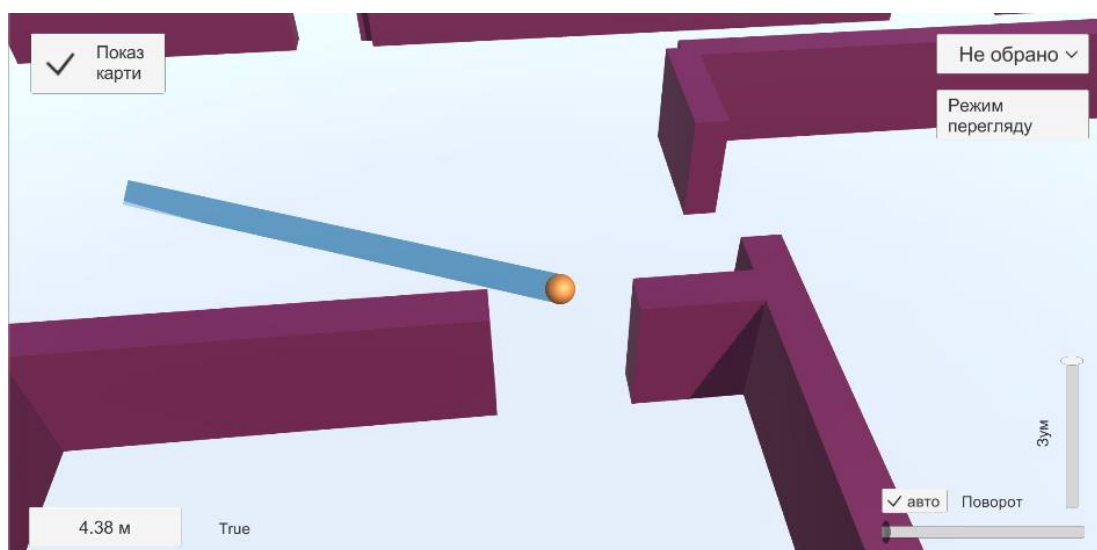


Рисунок 4.4 — Обрана користувацька точка призначення

У лівому нижньому кутку інтерфейсу розміщений текст, що показує відстань до першої точки шляху та довжину всього побудованого шляху.

У правому нижньому куті розміщені елементи ручного управління станом камери, а саме слайдери повороту та масштабу. Зміна масштабу доступна завжди переміщенням слайдера або характерним жестом двома пальцями розширення чи звуження. Прапорець «Авто» відповідає за автоматичний поворот камери, синхронний з поворотом мобільного пристрою, він є увімкненим в початковому стані додатку. Вимкнувши автоповорот, людина отримує здатність вільно повертати камеру навколо її фокусу, тобто центру екрану.

За замовчуванням, камера прив'язана до сфери-показника позиції та слідує за нею. Також передбачена функція вимикання прив'язки та вільне переміщення по схемі приміщення жестами прокручування. Для цього потрібно поставити прапорець «Режим перегляду».

Можливі два випадки під час обрання кінцевого місця навігації:

— Обрана точка на тому ж поверсі, де була проведена процедура синхронізації. В такому разі відразу показується шлях, слідуючи якому, можна прийти до точки призначення.

— Бажана кімната знаходиться на іншому поверсі. Тоді процес навігації ділиться на 3 етапи: перехід до найближчих сходів, що ведуть на потрібний поверх, підйом та повторна синхронізація з маркером, навігація до потрібного місця як у випадку 1.

У мобільному додатку також присутній режим доповненої реальності, що вмикається зняттям «Показ карти». В ньому показується зображення з камери, на яке накладаються віртуальні об'єкти, що інструктують відвідувачів про напрямок до обраного предмету чи кімнати.

Висновки до розділу

Отже, в розділі описані системні вимоги та порядок розгортання розробленого програмного продукту. Розглянуто сценарій роботи з мобільним додатком, порядок користування системою, її основні функції.

5 РОЗРОБКА СТАРТАП-ПРОЕКТУ

Мобільний додаток, що розробляється, підходить для створення стартап-проекту на його основі. В даному розділі описуються та обґрунтовуються складові стартапу, необхідні для планування та виходу на ринок.

5.1 Опис ідеї продукту

Усі проекти починаються з ідеї. Таблиця 5.1 наводить опис ідеї стартап-проекту.

Таблиця 5.1. Опис ідеї мобільного додатку

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Мобільний додаток внутрішньої навігації для смартфонів під управлінням ОС Android та iOS з елементами доповненої реальності.	Навігація в навчальних приміщеннях закладів освіти (школи, коледжі, університети).	Дозволяє шукати аудиторії. Особливо актуально для нових студентів та гостей під час подій, які часто проходять в корпусах університетів.
	Навігація в великих публічних робочих приміщеннях, наприклад, офісний центр, аеропорт.	Для власника приміщення відкриваються рекламні та моніторингові можливості. Працівникам та відвідувачам програма надає інструкції щодо переміщень. Також є можливим відображення метаданих.

У таблиці наведено зміст ідеї стартап-проекту, основні способи використання продукту та вигоди, які користувачі можуть отримати в тому чи іншому випадку. Мобільний додаток повинен бути персоналізований під кожний конкретний випадок

застосування. Він надає можливості перегляду схеми, знаходження кімнат та речей та інші сервіси на основі знань про позицію користувача.

Для визначення доцільності реалізації проекту варто визначити техніко-економічні характеристики програмного продукту.

Таблиця 5.2. Визначення переваг та недоліків проекту

№ п/п	Техніко- економічні характери- стики ідеї	(Потенційні) товари/концепції конкурентів				W (слабка сторона)	N (нейтральна)	S (сильна сторона)
		Розроблюваний проект	mobiDev	infSoft	IndoorAtlas			
1	Точність	середня	середня	висока	середня		+	
2	Надійність	низька	низька	висока	висока	+		
3	Собівартість	низька	середня	висока	висока			+
4	Використання додаткового обладнання	Ні	Ні	Так	Так			+
5	Кросплатфор- менність	Так	Так	Так	Так		+	
6	Використання доповненої реальності	Так	Так	Ні	Ні			+

Розуміння переваг та недоліків є основою для прийняття рішення щодо конкурентоспроможності стартап-проекту.

Основними перевагами програмного продукту є низька собівартість (по суті, тільки витрати на штат розробників) та відсутність додаткового обладнання, яке в інших випадках необхідно закуповувати, встановлювати та підтримувати. Всі розглянуті засоби є кросплатформенними, проте на деяких платформах існують обмеження. Так, наприклад, неможливо отримати дані про силу Wi-Fi сигналу в

пристроях Apple (iPhone/iPad). Це накладає певні обмеження на точність отриманих результатів. Розроблений додаток не містить функціоналу, який був би недоступним на деяких платформах.

Найбільшим недоліком програмного продукту є похибка, що накопичується. Це є платою за відсутність обладнання та є алгоритмічним недоліком методу. Бачимо, що прототип mobiDev також підвержений йому.

В цілому, мобільний додаток можна вважати конкурентоспроможним.

5.2 Технологічний аудит

У підрозділі потрібно провести аудит технологій, якими можна скористатись при розробці програмного продукту, необхідність розробки власних технологій.

Таблиця 5.3. Технологічний аудит проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Технології позиціонування	ARCore	Наявна	Доступна
		ARKit	Наявна	Доступна (тільки iOS)
		Vuforia	Наявна	Платна (100\$/міс)
2	Використання маркерів для синхронізації	QR-Code	Наявна	Доступна
		ArUco	Наявна	Доступна
Обрані технології реалізації ідеї проекту: ARCore, ArUco				

Таким чином, проект буде виконано з використанням технологій Google ARCore та маркерів ArUco. Вибір ARCore обґрунтований мультиплатформовістю, підтримкою рушія Unity та тим, що вона є безкоштовною. Альтернатива у вигляді ARKit виключена через обмеження ринку збуту пристроями Apple. Vuforia є платною, що не співпадає з принципом зменшення видатків.

Для маркерів обрано підхід ArUco через доступність словників та простоту реалізації генерації. QR коди не підходять так як є занадто деталізованими та складними для розпізнавання.

5.3 Аналіз ринкових можливостей

Вивчення ринкових можливостей та потенційних загроз проекту допомагає спланувати напрями зростання стартапу із урахуванням стану ринкового середовища, пропозицій конкурентів та потреб потенційних клієнтів. Спочатку проводимо аналіз попиту: стан ринку, динаміка, вимоги, що можуть вплинути на строки розробки.

Таблиця 5.4. Характеристики потенційного ринку стартап-проекту

№ п/п	Показники стану ринку	Характеристика
1	Кількість головних гравців, од.	Біля 5
2	Загальний обсяг продаж, грн/ум. од.	25 000 грн за ум. од.
3	Динаміка ринку (якісна оцінка)	Очікується зростання
4	Наявність обмежень для входу (вказати характер обмежень)	Немає
5	Специфічні вимоги до стандартизації та сертифікації	Немає
6	Середня норма рентабельності в галузі (або по ринку), %	15%-20%

Ринок внутрішньої навігації є досить молодим. Йому прогнозують значний ріст до 2025 року, припускають що його вартість складе приблизно 35 млрд доларів. Кількість головних гравців є незначною, більшість з них працюють з Bluetooth маяками, Wi-Fi, RFID, Ultra-Wideband та іншими радіотехнологіями.

Норма рентабельності є вищою за банківський відсоток вкладення. Це, а також відсутність явних перешкод роблять ринок IPIN гарним вибором для інвестицій.

Таблиця 5.5. Потенційні клієнти стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія	Відмінність у поведінці різних потенційних цілових груп клієнтів	Вимоги споживачів до товару
1	Можливість швидко і зручно здійснювати навігацію будівлею	Учні, студенти	Низька платоспроможність	Зручність інтерфейсу, швидкість роботи, можливість швидко дістатися до аудиторії
2	Можливість швидко і зручно здійснювати навігацію будівлею	Працівники офісів, клієнти аеропортів тощо	Більша вимогливість, можливість реалізації комерційних інтересів власника	Покращена точність (через те, що будівлі подібного характеру зазвичай більші)

Таким чином, зазначені категорії цільової аудиторії на момент виходу на ринок. З розвитком стартапу можливим є освоєння нових галузей, де придатна внутрішня навігація, що призведе до потреби адаптації продукту під нове використання. Початкове застосування та тестування програми стартап-проекту можливе в навчальних приміщеннях, наприклад, в навчальному корпусі №5 НТУУ «КПІ ім. Ігоря Сікорського». Після цього можна переходити на інші типи будівель з комерціалізацією. Для цього потрібна сильна маркетингова компанія.

Проаналізуємо фактори, що сприяють або зупиняють впровадження проекту.

Таблиця 5.6. Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Конкуренція	Розширення лідерів ринку на візуальні системи	— вихід; — продаж; — створити нові переваги, яких немає в жодного конкурента.
2	Ризик технології	Навігація без використання додаткового обладнання згодом може бути недостатньо точною для певного прошарку клієнтів	Підвищення точності за рахунок використання інноваційних систем або опціонального синтезу технологій (Wi-Fi/Bluetooth тощо)
3	Ліцензування	Використані програмні засоби можуть стати платними	Абстрагувати використання фреймворків, передбачити легку підміну реалізації
4	Зменшення користувачів додатку	Користувачам може набриднути/не подобатись користування програмою	Розробляти нові функції, оновлювати інтерфейс, ввести соціальний аспект/змагання в навігацію

У таблиці наведено основні загрози, що можуть запобігти ринковому впровадженню стартап-проекта. Найбільшим з них є небезпека виходу ІТ-гігантів на цей ринок, що є можливим, так як передбачається надзвичайний ріст його вартості та він зараз тільки розвивається. Цілком ймовірно, що компанії Google, Microsoft та інші вже зараз працюють на створенням аналогів, а з ними дуже важко конкурувати. Можливими виходами з такого становища є розробка власних «нечесних» переваг

(англ. unfair advantage), які неможливо скопіювати чи підробити, або просто продати компанію в разі зацікавленості зі сторони лідерів.

Меншим, проте все ще суттєвим, є ризик технології. Використані технології можуть виявитися недостатньо точними для клієнтів. Одним з рішень є додавання опціонального режиму синтезу технологій, в якому для покращення точності будуть вмикатись традиційні методи позиціонування (Wi-Fi, Bluetooth beacons, Ultra Wideband).

Найменшими по значимості є ризики ліцензування та відтоку користувачів через незадоволеність якістю програми.

Навпаки, деякі події можуть сприяти розвитку програми (таблиця 5.7).

Таблиця 5.7. Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Зростання купівельних можливостей клієнтів	Наразі є багато компаній, які хотіли б вбудувати систему навігації, проте це їм не по кишені; поява держфінансування	Постійно рекламувати та пропонувати продукт
2	Поява нових закладів освіти, офісних центрів або розширення існуючих	При зростанні кількості потенційних покупців (закладів, яким може знадобитися продукт), збільшиться попит на продукт	Більш агресивна маркетингова компанія
3	Підрив довіри до конкурентів	Зафіксовано витік приватної інформації про переміщення в конкурента	Інвестувати ресурси в покращення захисту даних
4	Збільшення відсотку пристроїв, що підтримуються	Наразі не всі смартфони підтримуються, проте цей відсоток обов'язково буде рости з часом	Акцентувати на цьому увагу при продажу

Отже, наведені фактори, що можуть посприяти збільшенню популярності стартап-проекту в порядку спадання значущості. Перші два з них є суттєвими. По-перше, може з'явитись потреба в подібних системах з боку держави. По-друге, в комерційних застосуваннях можливо буде досягнутий лавиноподібний ефект, коли власники намагаються не відставати від конкурентів.

Не виключається і покращення позицій продукту внаслідок помилок конкурентів. Одним з них може бути витік чутливої інформації про переміщення. Користувачі надзвичайно погано сприймають такі новини, адже нікому не подобається, що за ними стежать. Тому потрібно дуже обережно поводитись з даними телеметрії.

Також сприяючим фактором є зростаючий відсоток пристроїв, які підтримують ARCore. Поки що ARCore підтримують близько 500 млн пристроїв.

Визначимо загальні риси конкуренції на ринку (таблиця 5.8).

Таблиця 5.8. Ступеневий аналіз ринкової конкуренції

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства
Тип конкуренції — олігополія	Декілька сильних конкурентів	Враховувати цінову політику основних конкурентів
Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства
За рівнем конкурентної боротьби — глобальна	Ринок є міжнародним, в Україні представлений зовсім слабо	Варто розробляти продукт з підтримкою інтернаціоналізації
За галузевою ознакою — міжгалузева	Додаток може застосовуватись в різних галузях людської діяльності	Можна завоювати ринок в конкретній галузі

Таблиця 5.8 (продовження)

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства
Конкуренція за видами товарів — товарно-видова	Конкуренція між продуктами, які надають схожі можливості	Враховувати недоліки в продуктах конкурентів
За характером конкурентних переваг — нецінова	Продукти занадто різні, щоб змагатись за рахунок ціни	Оптимізувати вартість розробки та підтримки
За інтенсивністю — не марочна	Продукти не набувають настільки широкого розповсюдження, щоб бренд мав значення	-

У таблиці наведені основні характеристики ринку внутрішнього позиціонування та навігації. Ринок є дуже молодим, проте на ньому вже встановилися декілька сильних лідерів, тому тип конкуренції — олігополія. В результаті пошуку конкурентів, не було знайдено жодного, який працював би виключно в Україні. Частково українською є лише компанія mobiDev. Саме тому варто позиціонувати продукт як глобальний і підтримувати інтернаціоналізацію.

Продукти інших компаній надають схожий функціонал, проте різний рівень деталізації, налаштування під клієнта та підтримки, і не можуть конкурувати лише за рахунок ціни. Тут працює правило «кращий продукт за менші гроші». Конкуренція не є марочною.

Проведемо аналіз конкуренції за методикою 5 сил М. Портера. Згідно до неї, вирізняються п'ять сил, що задають рівень конкуренції: замінники, потенційні гравці, постачальники, споживачі та прямі конкуренти. За словами автора, модель п'яти сил варто застосовувати на макроекономічному рівні для визначення місця компанії в галузі. Такий аналіз може застосовуватись при виконанні умов: споживачі,

конкуренти та постачальники не є пов'язаними між собою, ціна визначається перевагами продукту, ринок є досить стабільним та дозволяє його учасникам планувати роботу та реакцію на дії конкурентів [46]. Перші дві сили описані в таблиці 5.9.

Таблиця 5.9. Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти
	Навести перелік прямих конкурентів	Визначити бар'єри входження в ринок
Висновки	Знайдено 5 конкурентів. Основні: Mobidev, HERE Indoor, NavVis. Інтенсивність боротьби низька	Строки виходу на ринок — приблизно 1 рік. Бар'єри — технологічні, розмір капіталовкладень. Потенційні конкуренти — ІТ-гіганти

Конкуренція хоч і присутня, проте досить слабка. Немає жодної компанії, яка активно використовувала би підхід візуально-інерційної навігації. Строки виходу складають від одного року в залежності від обраного підходу. Бар'єри — розмір капіталовкладень та технологічні.

Потенційними конкурентами є ІТ-гіганти, що можуть вирішити поширювати свій вплив і на ринок IPIN.

Наступна таблиця (5.10) розглядає інші сили в моделі Портера.

Таблиця 5.10. Аналіз конкуренції в галузі за М. Портером — інші складові

Складові аналізу	Постачальники	Клієнти	Товари-замінники
	Визначити фактори сили постачальників	Визначити фактори сили споживачів	Фактори загроз з боку замінників
Висновки	Постачальники відсутні	Продуктова диференціація, змінні витрати	Загроза розробки конкурентами більш точного та дешевшого аналогу

У нашому проекті постачальники відсутні. Проте в конкурентів вони наявні, наприклад, поставка Bluetooth маяків. Так як в проекті немає переваг, які неможливо підробити, існує загроза розробки замінника.

Перелік факторів конкурентоспроможності вказаний в таблиці 5.11.

Таблиця 5.11. Аргументація факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування
1	Ціновий	Передбачається випуск товару з ціною, нижчою ніж у конкурентів
2	Підтримки	В разі зміни оточення додаток продовжить роботу
3	Інфраструктурний	Для роботи програми не потрібне розгортання складної інфраструктури

Основною конкурентною перевагою є відсутність інфраструктури. Іншим фактором конкурентоспроможності є цінова політика — рішення має бути дешевшим, ніж в конкурентів. Плюсом є відсутність необхідності сканувати приміщення ще раз через деякий проміжок часу, як потрібно робити в разі застосування карт Wi-Fi та магнітного поля. Поки значно не зміниться планування поверху, додаток буде працювати.

Порівнюючи наш додаток та зазначені конкурентні рішення, можемо оцінити фактори конкурентоспроможності (таблиця 5.12).

Таблиця 5.12. Порівняння переваг та недоліків проектів на ринку

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з розроблюваним продуктом						
			-3	-2	-1	0	1	2	3
1	Ціновий	19	+						
2	Підтримки	15		+					
3	Інфраструктурний	17		+					

Найбільш переконливим фактором виявився ціновий — 19 балів. Найменше балів набрав фактор підтримки.

Виділимо сильні, слабкі сторони стартап-проекту, а також нереалізовані можливості та загрози існуванню компанії (таблиця 5.13). Такий метод називається SWOT-аналізом. Його назва походить від 4 сторін, що піддаються детальному розгляду (Strength, Weaknesses, Opportunities, Threats).

Таблиця 5.13. SWOT-аналіз стартапу

Сильні сторони: не потрібно покриття зони додатковими пристроями, AR-навігація, простота, підтримка метаданих	Слабкі сторони: Необхідність синхронізації позиції, накопичення похибки
Можливості: Опціональне використання додаткових технологій, розробка SDK, нові типи метаданих, покращення роботи між поверхами	Загрози: вихід на ринок великих компаній, що володіють значними ресурсами (Google, Microsoft), розробка іншою компанією універсальної точної системи позиціонування

SWOT-аналіз надає інформацію про можливості стартап-проекту щодо виходу на ринок. Можна підвести висновок, що існують значні загрози в майбутньому, проте додаток має сильні сторони, які в разі успішної реалізації можуть вивести компанію в лідери ринку.

Проаналізуємо альтернативні рішення розробки продукту та його впровадження (таблиця 5.14).

Таблиця 5.14. Альтернативи впровадження на ринок

№ п/п	Альтернатива	Ймовірність отримання ресурсів	Строки реалізації, роки
1.	Розробка мобільного додатку навігації з використанням радіопристроїв	50%	1,5

Таблиця 5.14 (продовження)

№ п/п	Альтернатива	Ймовірність отримання ресурсів	Строки реалізації, роки
2.	Розробка системи навігації з використанням тільки комп'ютерного зору	80%	1

Розробка традиційної системи з радіопристроями займає більше часу та зустріне більше опору при впровадженні на ринок, бо підхід є старим та часто застосовуваним, а тому конкурувати дуже важко. Розробка системи навігації з використанням комп'ютерного зору та доповненою реальністю є більш привабливою альтернативою.

5.4 Розробка ринкової стратегії

Для розробки ринкової стратегії обов'язково потрібно визначити цільові групи клієнтів. Таблиця 5.15 описує групи споживачів та їх особливості.

Таблиця 5.15. Обґрунтування вибору груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Заклади освіти	+	Середній	Практично немає	Середня
2	Комерційні центри торгівлі	+	Високий	Вище середнього	Вище середнього

Таблиця 5.15 (продовження)

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
3	Аеропорти, вокзали	+	Вище середнього	Висока	Складно
Які цільові групи обрано: 1, 2					

Бачимо, що найвища конкуренція спостерігається в області великих публічних приміщень, де дуже великий потік людей. Такі компанії мають достатні прибутки, щоб організувати розробку системи навігації. Проте на початку впровадження має сенс націлитись на групи користувачів, які тільки починають застосовувати IPS.

Всі проаналізовані групи споживачів готові сприйняти продукт та зацікавлені в перевагах, які він надає. Справа лише в фінансовій частині, складності розгортання та підтримки такої системи.

Таким чином, обрано цільові групи 1,2: комерційні та освітні заклади.

Визначимо базову стратегію розвитку (таблиця 5.16).

Таблиця 5.16. Вибір базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1	Розробка системи навігації з використанням тільки комп'ютерного зору	Стратегія диференційо ваного маркетингу	Функціонал, Вартість, Не потребує додаткового обладнання	диференціація

У якості стратегії охоплення ринку обрана стратегія диференційованого маркетингу. Вимоги до системи навігації в різних галузях сильно відрізняються. Тому складно буде організувати маркетингову компанію, яка зацікавила б всіх. Потрібно акцентувати увагу певного сегменту ринку на тому, що потрібно конкретно цим споживачам.

Вирізняють три базові стратегії розвитку, що одна від одної відрізняються типом конкурентної переваги, що покладена в основу розвитку компанії та ступенем охоплення ринку. Стратегія лідерства по витратам передбачає оптимізацію ціни за рахунок можливостей по збуту великих партій товару. Стратегія диференціації зосереджена на наданні покупцям нових властивостей продукту, за які вони готові платити більше. Третьою є стратегія спеціалізації, що заохочує концентрацію на потребах одного сегменту ринку, а не всіх. Кожна з них має власні переваги та недоліки. Концепція економії не підходить для високотехнологічних стартапів, тому була відкинута з розгляду, а стратегія спеціалізації занадто обмежує розвиток бізнесу, бо на даний момент клієнтів не так багато.

Обрана базова стратегія розвитку — диференціація, що передбачає розробку інноваційних рішень та ноу-хау. Тільки так компанія зможе залишитись на плаву.

Наступним етапом оцінки є визначення стратегії конкурентної поведінки (таблиця 5.17).

Таблиця 5.17. Вибір стратегії конкурентної боротьби

№ п/п	Чи є проект «першопрохід цем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
1	Ні	Шукати нових споживачів	Так, можливості навігації між поверхами за допомогою барометра (конкурент 2)	Стратегія наслідування лідеру

Проект, що розроблюється не є абсолютно новим на ринку. Так як ринок IPIN активно розвивається, легше буде шукати нових клієнтів, ніж відбирати існуючих.

Стратегія наслідування лідеру підходить в якості стратегії конкурентної поведінки. Бути лідером — надзвичайно дорого. Проте перебуваючи на 2-3 місці можна все ще отримувати гарні прибутки.

5.5 Створення маркетингової програми

У процесі формування маркетингової програми надзвичайно важливим є виділення вигоди, яку може запропонувати проект для задоволення конкретних потреб. Таблиця 5.18 наводить існуючі та потенційні ключові переваги, що повинні увійти до маркетингової програми.

Таблиця 5.18. Ключові переваги концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1.	Функціонал	Режим доповненої реальності	Наявність режиму, якого практично немає ні в одного з конкурентів
2.	Вартість	Вигідна ціна для покупців	Нижча ціна, ніж у більшості конкурентів
3.	Залежність системи від дод. пристроїв	Не потребує додаткового обладнання	Потрібен тільки смартфон

Наступним кроком є розробка трирівневої моделі товару (таблиця 5.19).

Таблиця 5.19. Трирівнева модель товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Допомагає знаходити потрібні речі, зони та кімнати в приміщенні, застосовуючи лише смартфон з підтримкою режиму доповненої реальності		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	Низька ціна	Нм	Тх
	Не потребує додаткових пристроїв		
	Зручний інтерфейс		
	Якість: ручне тестування, автоматизовані тести		
	Маркування: відсутнє		
	Компанія: «Tecinnav» (Technologies of indoor navigation)		
III. Товар із підкріпленням	Продаж товару за підпискою; постійна підтримка, вдосконалення та налаштування придбаного додатку		
За рахунок чого потенційний товар буде захищено від копіювання: обфускація коду			

Компанія називається «Tecinnav». Продуктом є мобільний додаток внутрішньої навігації. Якість буде перевірятися вручну та автоматизованими модульними та інтеграційними тестами.

Товар буде продаватися за моделлю підписки. Продавець зобов'язується підтримувати конкретне застосування, налаштовувати, розширювати його, розробляти новий функціонал в додатку. Потенційний товар буде захищено шляхом обфускації коду.

Експертним методом визначаються верхня та нижня межі встановлення цін на продукт (таблиця 5.20).

Таблиця 5.20. Межі встановлення цін

№ п/п	Рівень цін на товари- замінники	Рівень цін на товари- аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар
1.	30000	40000	350000	20000-25000

Для забезпечення переваги в ціновій політиці продукт має бути дешевшим, ніж аналоги. Був проведений аналіз потенційних місячних витрат компанії, включаючи маркетинг. На основі нього можна встановити вартість підписки — близько \$1000. Точка беззбитковості складає 11 проданих ліцензій.

Важливим є схема збуту товару. Таблиця 5.21 описує таку систему.

Таблиця 5.21. Опис системи збуту товару

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1.	Одноразова купівля та щомісячна підписка	Продаж	0 (напрямую)	Власна

Таким чином, оптимальним є прямий продаж та маркетинг. Клієнтів мало, а контракти цінні. Система збуту організована на основі щомісячної ліцензії на використання продукту.

Наступна таблиця (5.22) розглядає маркетингові комунікації.

Таблиця 5.22. Маркетингові комунікації стартап-проекту

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення
1.	Завантаження додатку зі спеціальних магазинів додатків. Посилання подається у вигляді QR- коду.	Інтернет	Безінфраструктурн ість, режим доповненої реальності, ціна, безпека користувацьких даних	Зацікавити потенційних користувачів; довести, що розроблений додаток економить час та допомагає спростити процес навігації.

Отже, розроблено стратегію маркетингу. Головним каналом комунікації є Інтернет. Є доцільним прямий маркетинг. Кінцеві користувачі можуть завантажити додаток, перейшовши за посиланням, що розміщене в цільовому приміщенні навігації. За можливості, потрібно використовувати маркетингові можливості замовника, так як йому також вигідно, щоб про мобільний додаток дізнались якомога більше відвідувачів.

Ключовими перевагами, які потрібно рекламувати кінцевим користувачам є зручність в користуванні, режим доповненої реальності, прискорення процесу навігації.

Висновки до розділу

Таким чином, у розділі досліджується доцільність виведення на ринок результату розробки даної та інших пов'язаних магістерських робіт — мобільного додатку навігації з використанням доповненої реальності. Продукт є цікавим як

власникам приміщенням, що хочуть оптимізувати процеси, так і кінцевим користувачам, що хочуть якнайшвидше знайти цікаві їм об'єкти.

У розділі значна увага присвячена визначенню слабких та сильних сторін продукту, що можуть впливати на його здатність конкурувати з іншими компаніями на ринку внутрішнього позиціонування та навігації (IPIN). Було проведено технологічний аудит проекту та обґрунтована доцільність вибору ARCore та маркерів ARUco, ігрового рушія Unity.

SWOT-аналіз та інші дослідження конкуренції показали, що розроблений продукт після незначного вдосконалення може бути виведений на ринок. Основними цільовими групами споживачів обрано навчальні заклади та комерційні центри.

Головним ризиком є висока ймовірність виходу на ринок IT-гігантів, які мають в своєму розпорядженні багато ресурсів. Тому доцільно розробляти продукт максимально швидко, паралельно вдосконалюючи його.

Продукт буде розповсюджуватись через фірмові магазини додатків відповідних платформ, тобто через Google Play та Apple AppStore. Основним методом маркетингу є прямий, тобто особисті зустрічі з демонстрацією переваг, які компанія-замовник може отримати після реалізації проекту.

Експертним методом визначено, що оптимальна ціна місячної підписки має складати 20000-25000 грн. Така вартість забезпечить цінове домінування над конкурентами, водночас буде забезпечувати достатній прибуток компанії. В разі більшого за прогнозований попиту на послуги компанії, ціну можна буде зробити вищою, а отримані гроші спрямувати на дослідження інноваційних технологій, які мають стати основою розвитку проекту.

Для проекту потрібні інвестиції в розмірі 600 тис. грн. Термін окупності складає 1 рік, тобто інвестиції будуть повернуті за цей період.

Остаточний висновок — додаток придатний для комерціалізації, подальша розробка та розвиток є доцільними.

ВИСНОВКИ

Для систем внутрішньої навігації головним є точне визначення позиції мобільного пристрою. В роботі розглянуто підходи, техніки та технології визначення позиції в просторі. Були проаналізовані існуючі програмні засоби навігації мобільних пристроїв у приміщеннях та література по темі. Виявилось, що не існує ідеального рішення задачі позиціонування. Існуючі рішення або занадто дорогі для масового використання або недостатньо точні. Тому розробка нових способів навігації залишається актуальною темою досліджень.

Традиційними технологіями є маяки Bluetooth та Wi-Fi «відбитки». Вони хоч і працюють без накопичення похибок, проте надають відносно невисоку точність та потребують встановлення десятків спеціальних пристроїв чи витрати часу на запис карти сигналів.

Сучасні смартфони надають більші можливості щодо використання алгоритмів комп'ютерного зору для задач позиціонування, так як мають камери, що якісно знімають та велику обчислювальну потужність. Тому наразі активно досліджуються та розробляються системи навігації лише з використанням зображення.

Була розроблена підсистема визначення положення пристрою на основі поєднання сенсорів телефону та зображення з камери в складі мобільного додатку навігації з використанням доповненої реальності. Застосування маркерів ArUco для початкової синхронізації довело свою працездатність та є придатним для подібних систем. Розгортання системи є максимально простим та включає лише друк та розташування постерів, що містять посилання на завантаження додатку та маркер синхронізації. Це і є основною перевагою перед існуючими програмними системами. Завдяки такому спрощенню система є набагато дешевшою.

Для реалізації проекту обрано інструмент Unity для розробки кросплатформових графічних додатків, що фактично є стандартом в сфері програм доповненої реальності. Логіку запрограмовано мовою C# в середовищі Visual Studio.

Платформа ARCore значно спростила процес імплементації програмного продукту, так як надає багато спеціальних інтерфейсів: оцінка освітлення, відстеження переміщення пристрою, розпізнавання обличч та зображень.

Додаток та розроблений модуль були протестовані вручну та показали задовільну точність, якої достатньо для навігації в корпусі навчального закладу. Засіб ARCore виявився обмежувальним фактором в розробленому продукті, так як не дозволяє завантажувати раніше записану точкову карту в пам'ять для потреб локалізації. Проте нам не відомі аналоги, які дозволяли б це зробити. Можливим вирішенням проблеми є ручна інтеграція візуально-інерційної одометрії в мобільний додаток з отриманням низькорівневого доступу до структур даних методу. Тому в програмній системі залишається потенціал до вдосконалення та покращення точності та інших функціональних якостей.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Mautz R. Indoor positioning technologies : дис. докт. / Mautz Rainer — Zurich, 2012. — 129 с.
2. Sonnenberg G. Radar and electronic navigation / G. Sonnenberg. — Cambridge: Butterworth & Co, 1988. — (6).
3. Hofmann-Wellenhof B. Navigation: Principles of positioning and guidance / B. Hofmann-Wellenhof, K. Legat, M. Wieser. — New York: Springer, 2011.
4. Prasad R. Applied satellite navigation using GPS, GALILEO, and augmentation systems / R. Prasad, M. Ruggieri. — Boston: Artech, 2005.
5. Gu Y. A survey of indoor positioning systems for wireless personal networks / Y. Gu, A. Lo, I. Niemegeers. // IEEE Communications Surveys & Tutorials. — 2009. — №11. — С. 13—32.
6. Lemmens M. Indoor Positioning. Technologies, features and prospects. / M. Lemmens. — 2013.
7. Nuaimi K. A survey of indoor positioning systems and algorithms / K. Nuaimi, H. Kamel. // International Conference on Innovations in Information Technology. — 2011. — С. 185—190.
8. The Active Badge Location System / R. Want, A. Hopper, V. Falcao, J. Gibbons. // ACM Transactions on Information Systems. — 1992. — №10. — С. 91—102.
9. Priyantha N. The Cricket Location-Support system / N. Priyantha, A. Chakraborty, H. Balakrishnan. // ACM MOBICOM. — 2000. — №6.
10. Magnetic position and orientation tracking system / F. H. Raab, E. B. Blood, T. O. Steiner, H. R. Jones. // IEEE Transactions on Aerospace and Electronic Systems. — 1979. — №15. — С. 709—718.
11. Sakpere W. A state-of-the-art survey of indoor positioning and navigation systems and technologies / W. Sakpere, M. Adeyeye-Oshin, N. Mlitwa. // South African Computer Journal. — 2017. — №29. — С. 145—197.

12. Mautz R. Survey of optical indoor positioning systems / R. Mautz, S. Tilch. // International Conference on Indoor Positioning and Indoor Navigation. — 2011. — №2011. — C. 1—7.
13. Raj R. C. QR code based navigation system for closed building using smart phones / R. C. Raj, S. B. Tolety, C. Immaculate. // International Mutli-Conference on Automation, Computing, Communication, Control and Compressed Sensing (iMac4s). — 2013. — C. 641—644.
14. Hile H. Positioning and orientation in indoor environments using camera phones / H. Hile, G. Borriello. // IEEE Computer Graphics and Applications. — 2008. — №28. — C. 32—39.
15. Kohoutek T. K. Real-time indoor positioning using range imaging sensors / T. K. Kohoutek, R. Mautz, A. Donaubauer. // Proceedings of the Real-Time Image and Video Processing. — 2010. — №7724. — C. 1—8.
16. Kim J. Vision-based location positioning using augmented reality for indoor navigation / J. Kim, H. Jun. // IEEE Transactions on Consumer Electronics. — 2008. — №54. — C. 954—962.
17. Werner M. Indoor positioning using smartphone camera / M. Werner, M. Kessel, C. Marouane. // International Conference on Indoor Positioning and Indoor Navigation. — 2011.
18. Indoor positioning and navigation with camera phones / A. Mulloni, D. Wagner, I. Barakonyi, D. Schmalstieg. // IEEE Pervasive Computing. — 2009. — №8. — C. 22 — 31.
19. A mobile indoor navigation system interface adapted to vision-based localization / [A. Möller, M. Kranz, R. Huitl та ін.]. // International Conference on Mobile and Ubiquitous Multimedia. — 2012. — №11. — C. 1—10.
20. Mountney P. Simultaneous Stereoscope Localization and Soft-Tissue Mapping for Minimal Invasive Surgery / P. Mountney. // MICCAI. Lecture Notes in Computer Science. — 2006. — №1. — C. 347—354.

21. Bailey T. Simultaneous Localisation and Mapping (SLAM): Part II / T. Bailey, H. F. Durrant-Whyte. // Robotics and Autonomous Systems (RAS). — 2006. — №13. — C. 108—117.
22. A review of recent developments in Simultaneous Localization and Mapping / G.Dissanayake, S. Huang, Z. Wang, R. Ranasinghe. // International Conference on Industrial and Information Systems. — 2011. — C. 477—482.
23. Engel J. LSD-SLAM: Large-Scale Direct Monocular SLAM / J. Engel, T. Schops, D. Cremers. // European Conference on Computer Vision (ECCV). — 2014.
24. Mur-Artal R. ORB-SLAM: A Versatile and Accurate Monocular SLAM System / R. Mur-Artal, J. Montiel, J. D. Tardós. // IEEE Transactions on Robotics. — 2015. — №31. — C. 1147—1163.
25. Davidson A. J. Real-time simultaneous localisation and mapping with a single camera / Andrew J. Davidson. // IEEE International Conference on Computer Vision. — 2003.
26. Longuet-Higgins C. H. A computer algorithm for reconstructing a scene from two projections / C. Hugh Longuet-Higgins. // Nature. — 1981. — №293. — C. 133—135.
27. Lu F. Globally consistent range scan alignment for environment mapping / F. Lu, E. Milios. // Autonomous Robots (AR). — 1997. — №4. — C. 333—349.
28. Bundle Adjustment — A Modern Synthesis / B.Triggs, P. McLauchlan, R. Hartley, A. Fitzgibbon. // Vision Algorithms: Theory and Practice. — 2000. — C. 298—375.
29. A General Framework for Graph Optimization. In Proceedings of the IEEE International Conference on Robotics and Automation / [R. Kummerle, G. Grisetti, H. Strasdat та ін.]. // Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). — 2011. — C. 3607—3613.
30. Mourikis A. I. A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation / A. I. Mourikis, S. I. Roumeliotis. // Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). — 2007. — C. 3565—3572.
31. Camera-IMU-based localization: Observability analysis and consistency improvement. / J.Hesch, D. Kottas, S. Bowman, S. Roumeliotis. // The International Journal of Robotics Research (IJRR). — 2014. — №33. — C. 182—201.

32. Nister D. Scalable Recognition with a Vocabulary Tree / D. Nister, H. Stewenius. // n Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). — 2006. — №2. — С. 2161—2168.
33. Scaramuzza D. Visual Odometry [Tutorial]. Part I: The First 30 Years and Fundamentals / D. Scaramuzza, F. Fraundorfer. // IEEE Robotics and Automation Magazine. — 2011. — №18. — С. 80—92.
34. Latif Y. Robust loop closing over time for pose graph slam / Y. Latif, C. Cadena, J. Neira. // The International Journal of Robotics Research (IJRR). — 2013. — №32. — С. 1611—1626.
35. Bundle adjustment in the large. / [S. Agarwal, N. Snavely, I. Simon та ін.]. // European Conference on Computer Vision (ECCV). — 2010. — С. 29—42.
36. Shen C. Interpolating and Approximating Implicit Surfaces from Polygon Soup / C. Shen, J. F. O'Brien, J. R. Shewchuk. // SIGGRAPH. — 2004. — С. 896—904.
37. SLAM++: Simultaneous Localisation and Mapping at the Level of Objects / [R. Salas-Moreno, R. Newcombe, H. Strasdat та ін.]. // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). — 2013. — С. 1352—1359.
38. Solutions for Real-Time Locating Systems (RTLS) by infsoft [Електронний ресурс] — Режим доступу до ресурсу: <https://www.infsoft.com/>.
39. NavVis | Home [Електронний ресурс] — Режим доступу до ресурсу: <https://www.navvis.com/>.
40. Huang H. A Survey of Mobile Indoor Navigation Systems / H. Huang, G. Gartner. // Cartography in Central and Eastern Europe: Springer. — 2010. — С. 305—319.
41. Romero-Ramirez F. Speeded up detection of squared fiducial markers / F. Romero-Ramirez, R. Munoz-Salinaz, R. Medina-Carnicer. // Image and Vision Computing. — 2018. — №76. — С. 38—47.
42. ARTag, AprilTag and CALTag Fiducial Marker Systems: Comparison in a Presence of Partial Marker Occlusion and Rotation / [A. Sagitov, K. Shabalina, L. Sabirova та ін.]. // International Conference on Informatics in Control, Automation and Robotics. — 2017. — №14. — С. 182—191.

43. Automatic generation and detection of highly reliable fiducial markers under occlusion / S.Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, M. J. Marín-Jiménez. // Pattern Recognition. — 2014. — №47. — С. 2280—2292.
44. Unity Real-Time Development Platform [Електронний ресурс] — Режим доступу до ресурсу: <https://unity3d.com/unity>.
45. ARCore — Google Developers [Електронний ресурс] — Режим доступу до ресурсу: <https://developers.google.com/ar>.
46. Розроблення стартап-проекту [Електронний ресурс] : Методичні рекомендації до виконання розділу магістерських дисертацій для студентів інженерних спеціальностей / За заг. ред. О.А. Гавриша. — Київ : НТУУ «КПІ», 2016. — 28 с.

ДОДАТОК А

Підсистема визначення позиції в просторі для мобільного додатку навігації з використанням доповненої реальності

Текст програми

УКР.НТУУ”КПІ ім. Ігоря Сікорського”_ТЕФ_АПЕПС_ТВ42135_19М 12-1

Аркушів 9

Київ — 2019

TrackingController.cs

```

using Assets.Scripts.UI;
using GoogleARCore;
using UnityEngine;
using UnityEngine.UI;

namespace Assets.Scripts
{
    public class TrackingController : MonoBehaviour
    {
        public Text CamPoseText;
        public CameraState CameraState;
        public GameObject CameraTarget;
        public PointerState PointerState;

        const int LostTrackingSleepTimeout = 15;

        public void Update()
        {
            QuitOnConnectionErrors();
            if (Session.Status == SessionStatus.LostTracking)
            {
                CamPoseText.text = UiConstants.LostTrackingMessage;

                Screen.sleepTimeout = LostTrackingSleepTimeout;
                return;
            }

            // Clear camPoseText if no error
            Screen.sleepTimeout = SleepTimeout.NeverSleep;
            CamPoseText.text = string.Empty;

            if (CameraTarget != null)
            {
                var pos =
                PointerState.GetPointerPosition(CameraState.RotationDrift);

                // We apply translation only in the XZ plane.
                CameraTarget.transform.position = new Vector3(pos.x,
                CameraTarget.transform.position.y, pos.z);
            }
        }

        private void QuitOnConnectionErrors()
        {
            // Do not update if ARCore is not tracking.
            if (Session.Status ==
            SessionStatus.ErrorSessionConfigurationNotSupported)
            {
                CamPoseText.text = "This device does not support ARCore.";
            }
        }
    }
}

```

```

        Application.Quit();
    }
    else if (Session.Status ==
SessionStatus.ErrorPermissionNotGranted)
    {
        CamPoseText.text = "Camera permission is needed to run this
application.";
        Application.Quit();
    }
    else if (Session.Status == SessionStatus.ErrorApkNotAvailable)
    {
        CamPoseText.text = "ARCore encountered a problem connecting.
Please start the app again.";
        Application.Quit();
    }
    }
}
}

```

ArucoGenerator.cs

```

using System;
using System.Collections.Generic;
using System.IO;
using System.Threading;
using Assets.Scripts.Map;
using GoogleARCore;
using Newtonsoft.Json;
using UnityEngine;

public class ArucoGenerator : MonoBehaviour
{
    public TextAsset ArucoDictionary;
    public string DictName;
    public int Size;
    public float MarkerPhysicalSize = 0.135f;
    public AugmentedImageDatabase ImageDatabase;

    private string _dictJson;
    private Dictionary<string, byte[][]> _dict;

    public void Start()
    {
        _dictJson = ArucoDictionary.text;
        _dict = JsonConvert.DeserializeObject<Dictionary<string,
byte[][]>>(_dictJson);
    }

    public Texture2D GenerateMarker(int markerId)
    {
        if (_dict == null)

```

```

        return null;

var bytes = _dict[DictName][markerId];

var bits = new List<Color>();
var bitsCount = Size * Size;

// Parse marker's bytes
foreach (var b in bytes)
{
    var start = bitsCount - bits.Count;
    for (var i = Math.Min(7, start - 1); i >= 0; i--)
    {
        bits.Add(((b >> i) & 1) > 0 ? Color.white : Color.black);
    }
}

var texture = new Texture2D(Size + 4, Size + 4, TextureFormat.RGB24,
false);
for (int i = 0; i < Size + 4; i++)
{
    for (int j = 0; j < Size + 4; j++)
    {
        if(i == 0 || i == (Size + 3) || j == 0 || j == (Size + 3))
            texture.SetPixel(i, j, Color.white);
        else if (i == 1 && j > 0 && j < Size + 3
            || i == (Size + 2) && j > 0 && j < Size + 3
            || j == 1 && i > 0 && i < Size + 3
            || j == (Size + 2) && i > 0 && i < Size + 3)
            texture.SetPixel(i, j, Color.black);
    }
}

for (int i = 0; i < Size; i++)
{
    for (int j = 0; j < Size; j++)
    {
        texture.SetPixel(j + 2, i + 2, bits[(Size - 1 - i) * Size +
j]);
    }
}

return TextureScaler.scaled(texture, 1024, 1024, FilterMode.Point);
}

public void AddToImageDatabase(int markerId)
{
    var markerTexture = GenerateMarker(markerId);
    //var tempFile = Path.GetTempFileName();
    //File.WriteAllBytes(tempFile, markerTexture.EncodeToPNG());
    if (markerTexture != null)

```

```

        {
            var index = ImageDatabase.AddImage($"Marker{markerId}",
markerTexture, MarkerPhysicalSize);
            Debug.Log($"Added an image, id = {index}");
        }
    }
}

```

MarkersController.cs

```

using System.Collections.Generic;
using System.Linq;
using Assets.Scripts.Map;
using GoogleARCore;
using UnityEngine;
using UnityEngine.UI;

namespace Assets.Scripts.Markers
{
    public class MarkersController : MonoBehaviour
    {
        public Text LogText;
        public GameObject Device;
        public Camera Camera;
        public CameraState CameraState;
        public PointerState PointerState;
        public GameObject TrackerSphere;
        public DynamicMapOptions Options;
        public SessionResetter Resetter;

        private readonly Dictionary<int, Anchor> _anchors = new
Dictionary<int, Anchor>();
        private readonly List<AugmentedImage> _tempAugmentedImages = new
List<AugmentedImage>();
        private bool _fullyTracking = false;

        void Update()
        {
            // Get updated augmented images for this frame.
            Session.GetTrackables(_tempAugmentedImages,
TrackableQueryFilter.Updated);

            // Create visualizers and anchors for updated augmented images
that are tracking and do
            // not previously have a visualizer. Remove visualizers for
stopped images.
            foreach (var image in _tempAugmentedImages)
            {
                _anchors.TryGetValue(image.DatabaseIndex, out var anchor);
                if (image.TrackingState == TrackingState.Tracking

```



```

        && image.TrackingMethod ==
AugmentedImageTrackingMethod.FullTracking
        && anchor == null)
    {
        var id = int.Parse(image.Name.Replace("Marker",
string.Empty));
        var marker = Options.Markers.FirstOrDefault(m =>
m.MarkerId == id);
        if (marker != null)
        {
            LogText.text = $"Marker #{marker.MarkerId} is found";

            Options.CurrentScheme = marker.Scheme;

            //Optionally reset session
            //Resetter.StartSessionReset();
        }

        // Create an anchor to ensure that ARCore keeps tracking
this augmented image.
        anchor = image.CreateAnchor(image.CenterPose);
        _anchors.Add(image.DatabaseIndex, anchor);
    }
    else if (image.TrackingState == TrackingState.Tracking
        && image.TrackingMethod ==
AugmentedImageTrackingMethod.FullTracking
        && anchor != null)
    {
        var id = int.Parse(image.Name.Replace("Marker",
string.Empty));
        var marker = Options.Markers.FirstOrDefault(m =>
m.MarkerId == id);
        if (marker != null && !_fullyTracking)
        {
            LogText.text = $"Marker #{marker.MarkerId} is found";

            //Set rotation difference
            var markerYAngle = Vector3.SignedAngle(Vector3.right,
marker.transform.up, Vector3.up);
            var currentYAngle =
CameraState.GetOriginalCameraRotation().eulerAngles.y;
            CameraState.RotationDrift = Quaternion.Euler(0,
markerYAngle - currentYAngle + 90, 0);

            var spherePos = marker.gameObject.transform.position
+ marker.gameObject.transform.up * 0.3f;
            var originalPos =
PointerState.GetOriginalPointerPosition(CameraState.RotationDrift);
            PointerState.PositionDrift = new Vector3(spherePos.x,
originalPos.y, spherePos.z) - originalPos;

```



```

#else
    return rotationDrift * Quaternion.Euler(0, 180, 0) *
Frame.Pose.position;
#endif
}

public event PropertyChangedEventHandler PropertyChanged;

[NotifyPropertyChangedInvoker]
protected virtual void OnPropertyChanged([CallerMemberName] string
propertyName = null)
{
    PropertyChanged?.Invoke(this, new
PropertyChangedEventArgs(propertyName));
}
}
}

```

CameraState.cs

```

using System.ComponentModel;
using System.Runtime.CompilerServices;
using GoogleARCore;
using JetBrains.Annotations;
using UnityEngine;

namespace Assets.Scripts
{
    public class CameraState : MonoBehaviour, INotifyPropertyChanged
    {
        public Quaternion Rotation { get; set; } = Quaternion.identity;
        public Quaternion RotationDrift { get; set; } = Quaternion.identity;

        public bool AutoRotation { get; set; } = true;

        public Quaternion GetCameraRotation()
        {
            return GetOriginalCameraRotation() * RotationDrift;
        }

        public Quaternion GetOriginalCameraRotation()
        {
            return AutoRotation
                ? Frame.Pose.rotation
                : Rotation;
        }

        [NotifyPropertyChangedInvoker]
        protected virtual void OnPropertyChanged([CallerMemberName] string
propertyName = null)
        {

```

```
        PropertyChanged?.Invoke(this, new
PropertyChangedEventArgs(propertyName));
    }

    public event PropertyChangedEventHandler PropertyChanged;
}
}
```

ДОДАТОК Б

Підсистема визначення позиції в просторі для мобільного додатку навігації з
використанням доповненої реальності

Апробація

УКР.НТУУ"КПІ ім. Ігоря Сікорського"_ТЕФ_АПЕПС_TV42135_19М

Аркушів 4

Київ — 2019

Було подано тези щодо розробленого програмного продукту до XVII міжнародної науково-практичної конференції «Сучасні проблеми наукового забезпечення енергетики», яка проходила в Києві 23-26 квітня 2019 року. Тези надруковані в томі №2 на сторінці 95. Текст доповіді наведений нижче.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

СУЧАСНІ ПРОБЛЕМИ НАУКОВОГО ЗАБЕЗПЕЧЕННЯ ЕНЕРГЕТИКИ

Матеріали XVII Міжнародної
науково-практичної конференції
молодих вчених та студентів
м. Київ, 23-26 квітня 2019 року,

ТОМ 2



Київ- 2019

Рисунок Б.1 — Обкладинка тому №2 збірника тез

<i>МОСКАЛЕНКО Ю.В., аспірант</i>	
Машинне навчання для розв'язання логічних головоломок.	93
<i>БАРАНІЧЕНКО О.М., магістрант гр. ТВ-71мн</i>	
<i>Керівник - доц., к.т.н. Шаповалова С.І.</i>	
Веб-середовище для моделювання процесів міжагентної взаємодії в мережах Smart Grid.	94
<i>ШВАЙКА Д.А., магістрант гр. ТР-81мн</i>	
<i>Керівник - доц., к.ф.-м.н. Тарнавський Ю.А.</i>	
Використання техніки Structure from Motion в системі навігації.	95
<i>ХАРАБАР В.В., магістрант гр. ТВ-81мн</i>	
<i>Керівник - доц., к.т.н. Гагарін О.О.</i>	
Система оцінювання екологічних збитків у мережі АЗС.	96
<i>ОЛЕКСІЙ А.О., магістрант гр. ТВ-82</i>	
<i>Керівник - доц., к.т.н. Гагарін О.О.</i>	
Інтелектуальна система розпізнавання та передбачення намірів користувача.	97
<i>МЕЛЬНИЧЕНКО А.В., магістрант гр. ТВ-81мн</i>	
<i>Керівник - ст.викл., к.т.н. Шалденко О.В.</i>	
Проблема формування схеми замкнутого простору у системах внутрішньої навігації.	98
<i>МАРУНЯ А.В., магістрант гр. ТВ-81мн</i>	
<i>Керівник - доц., к.т.н. Гагарін О.О.</i>	
Застосування нейронних мереж в мобільних застосунках.	99
<i>МАРИЧ Т.І., магістрант гр. ТВ-81мн</i>	
<i>Керівник - доц., к.т.н. Шаповалова С.І.</i>	
Генерація елементів цифрового контенту на основі аналізу тексту.	100
<i>КРЮЧКОВСЬКА А.В., магістрант гр. ТВ-81мн</i>	
<i>Керівник - ст.викл., к.т.н. Шалденко О.В.</i>	
Програмний інструментарій виокремлення заданих об'єктів на зображенні.	101
<i>КРУТЛИК Д.С., магістрант гр. ТВ-81мн</i>	
<i>Керівник - доц., к.т.н. Шаповалова С.І.</i>	
Система розпізнавання жестів рук для людино-машинної взаємодії.	102
<i>КОНКІНА Н.С., магістрант гр. ТВ-81мн</i>	
<i>Керівник - ст.викл., к.т.н. Шалденко О.В.</i>	
Проблема вибору раціонального методу позиціювання користувача для системи навігації.	103
<i>ЗАРИЦЬКИЙ В.П., магістрант гр. ТВ-81мн</i>	
<i>Керівник - доц., к.т.н. Гагарін О.О.</i>	
Нейромережеве архітектурне рішення для обробки аудіосигналів.	104
<i>ВИТВИЦЬКИЙ Д.А., магістрант гр. ТВ-81мн</i>	
<i>Керівник - ст.викл., к.т.н. Мажара О.О.</i>	
Побудова сучасного веб-серверу на основі безсерверних технологій.	105
<i>БРУНЬКО П.В., магістрант гр. ТВ-81мн</i>	
<i>Керівник - доц., к.т.н. Шаповалова С.І.</i>	
Сегментація бур'янів на зображеннях з відеокамери наземного робота.	106
<i>СОФІЄНКО А.Ю., студент гр. ТР-52</i>	
<i>Керівник - доц., к.т.н. Шаповалова С.І.</i>	
Серверна частина системи функціонування реєстру інформаційних ресурсів.	107
<i>СОЛОМКІН Д.Г., студент гр. ЗПІ-ЗП-63</i>	
<i>Керівник - ст.викл. Гайдаржи В.І.</i>	

УДК 004.932.2

Магістрант 5 курсу, гр. ТВ-81мп Харабар В.В.

Доц., к.т.н. Гагарін О.О.

ВИКОРИСТАННЯ ТЕХНІКИ STRUCTURE FROM MOTION В СИСТЕМІ НАВІГАЦІЇ

За останні роки при вирішенні задач комп'ютерного зору, а саме створення тривимірних карт для потреб навігації, дослідники запропонували багато підходів до реалізації систем відновлення 3D структури тіл з набору зображень. У наведеному матеріалі пропонується варіант рішення такого класу задач з використанням техніки генерації структури з руху (англ. SfM — Structure from Motion).

Техніка SfM використовується для побудови тривимірної моделі простору у вигляді множини 3D-точок, що відповідають формі об'єктів, які розміщуються у цьому просторі.

Джерелом даних служать фотографії простору зсередини, які можливо достатньо просто отримати сучасними смартфонами. Основною ідеєю методу є те, що зображення одного й того ж об'єкту з різних сторін дозволяють відновити інформацію про об'єм. Унікальною особливістю SfM є те, що з комбінації попередньо створеної хмари точок простору та фотографії можливо дізнатись позицію та орієнтацію камери в просторі на момент зйомки, та вирішати на основі неї задачі навігації.

Загальна послідовність виокремлення тривимірних точок з набору фотографій є наступною [1]:

1. Вилучення ознак із зображення.
2. Знаходження схожих ознак шляхом порівняння пар зображень.
3. Геометрична перевірка правильності взаємного розташування камер.
4. Ущільнення хмари точок та створення полігональної моделі.

Реконструювання сцени у задачі навігації починається з вибору найкращої пари зображень. Потім до неї послідовно додаються всі інші зображення. Деякі з них можуть бути відкинуті з процесу, якщо їм неможливо знайти відповідні комплементарні фотографії.

Останній крок наведеної вище техніки займає досить багато часу, але мінімально впливає на точність визначення розташування камери (користувача) в навігаційній системі. Саме тому доцільно відмовитись від нього в циклічному процесі локалізації.

Існує багато засобів генерації тривимірної моделі з 2D-зображень. Деякі з них є комерційними веб-сервісами, як то Microsoft Photosynth, Autodesk 123D Catch та 3DF Zephyr, інші — з відкритим кодом та безкоштовні для використання. Найбільш популярними відкритими рішеннями є Bundler, VisualSFM та SFMToolkit.

Підхід має недоліки: необхідна велика кількість зображень та обчислювальних ресурсів для забезпечення прийнятної точності та складність отримання множини «правильних» фотографій зі значним перекриттям об'єктів. Як наслідок, на поточний момент, метод не можна застосовувати для дуже великих просторових об'єктів. Перевагами ж є широкий вибір алгоритмів та можливостей їх конфігурації, наявні оптимізовані реалізації алгоритмів на графічних прискорювачах, достатня точність локалізації на невеликих наборах даних.

Отже, виділені переваги та недоліки підходу дозволяють сподіватися на доцільність використання методу відновлення структури з руху у системах навігації з застосуванням комп'ютерного зору.

Перелік посилань:

1. Schonberger J. Structure-from-Motion Revisited / J. Schonberger, J. Frahm. // The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). — 2016